
DOS/V

DOS/V

テクニカル

Technical

リファレンス

Reference

マニュアル

Manual

芦達 剛／著

SOFTBANK BOOKS 

DOS/V

DOS/V

テクニカル

Technical

リファレンス

Reference

マニュアル

Manual

芦達 剛／著

SOFTBANK BOOKS 

本書中の商品名は一般に各社の登録商標です。
本文中に、TM、®マークは明記していません。

© 1993 本書のプログラムを含むすべての内容は著作権法上の保護を受けています。著者、
発行者の許諾を得ず、無断で複写、複製をすることは禁じられています。

はじめに

「本は一人で書けるものではない」

これまでの執筆活動の中で、今回ほどこのことを痛切に感じたことはありませんでした。他の業務と本書の入稿が重なってしまい、そのため度重なる原稿の遅れが生じたにも関わらず忍耐強く待っていただいたソフトバンクパソコン言語書籍編集部野沢編集長と編集部の藤山さん。しつこいまでの質問にも迅速に答えていただいた日本 IBM 社技術サポートの加来徹也氏。過密スケジュールの時間の合間を縫うようにして資料探しにご協力いただいた日本 IBM 社の岩本健一氏。

それから、こちらの無理な要求に、わざわざイレギュラーな方法を取ってまでご協力いただいた OADG(PC オープン・アーキテクチャ推進協議会)事務局の方々、そして休日を潰してまでの執筆にも文句もいわずに見守ってくれた女房…。なんと多くの方々に迷惑をおかけしたことか…。

そう考えると、本書は筆者が書いたものではなく、多くの人々の協力の成果といえるのかもしれない。ですから、本書が…、この成果が少しでも多くの人々のお役に立つことができれば、これほどうれしいことはありません。

本書を、本書の完成に寄与して下さったすべての方々に捧げたいと思います。

なお、本書で参考にさせていただいた資料は、日本 IBM 株式会社と OADG(PC オープン・アーキテクチャ推進協議会)のご協力によるものです。ここにお礼申し上げます。

1993 年 8 月

著者記す

本書の構成

本書は、DOS/V 対応アプリケーションを開発するためのテクニカル・リファレンス用マニュアルです。

DOS/V の登場により、「世界の標準機」といわれる IBM-PC でも日本語の使用が可能となりました。これにより、世界中のパーソナル・コンピュータ市場の中にあって唯一台風の目であった日本市場にも、海外の安価で高性能な IBM-PC 互換機が入り始め、今、日本のパーソナル・コンピュータ業界全体が大きな時代のうねりに巻き込まれようとしています。

しかし、日本では IBM-PC に対する認識はまだまだ低く、IBM-PC 自体に対するノウハウも皆無に等しい状況です。

さらに DOS/V は、今日の国際的なハードウェア環境の上に構築されたシステムであり、将来の多国語へのサポートも念頭に置いたシステムですので、従来の DOS にはなかった多くの複雑な問題も抱えています。

本書ではこれらの点を考慮して、よりよいアプリケーションを作成するための数々の方策を盛り込みました。

まず、1 点は「互換性の問題」です。「IBM-PC は世界中にただ 1 つのアーキテクチャしかないのだから、互換性は考慮しなくてもよいのに」と考えられる方がまだまだ多いと思いますが、これは大きな誤解です。

IBM-PC とその互換機にはハードウェア的に多くの亜流が存在していますが、これまでのソフトウェアはそれらをサポートしてきたために、それが表だっては見えなかっただけのことなのです(それがノウハウというものでしょう)。

たとえば BIOS を 1 つ取ってみても、IBM の PC-AT、PS/2、PS/55 ではその機能が異なる部分が多々あります。したがって PC-AT の資料を元にソフトウェアを作成しても、すべての DOS/V 対応機で完全に動作するとは限らないのです。NEC の PC-9801 でも、EGC(Enhanced Graphics Charger)を使用すると VM 以前の機種では動作しなくなりますが、それと同じことが IBM-PC にもいえるわけです。

そこで本書では互換性を重視し、解説を「すべての DOS/V 対応機で利用可能な機能」に限定しました。その結果、他の IBM-PC の技術書と比較すると本書の内容はずいぶん少なくなっていますが、これは「互換性の高いアプリケーション」を作成するために、あえて解説しなかった部分があるためです。

つまり、本書の解説を基準にアプリケーションを開発すれば、より多くの DOS/V 対応

機に対して安定動作が望める、ということです。

2 点目は互換レベルの問題です。DOS/V に限らず、アプリケーションを開発しているときに、「これはどちらの方法でも実現できるが、どちらがよりよい方法だろう」と判断に迷うときがあります。

そこで本書では、先に示した解説のほうが後の解説より、その互換性が高くなるように設定しました。これにより、本書を参照していただく段階から互換レベルの問題を意識的に考慮できるだけではなく、前半の章を参照していただくだけで DOS/V にあまり慣れていない方でも、開発を始めることができるという構成になっています。

これ以外にも、本書では読者の便を考えて何点かの工夫を盛り込みました。

- ・ 本書内での用語はできる限り統一していますが、これはより一般的な呼称と思われるものを採用しました。ですから、これらの用語は IBM の用語とも OADG の用語とも異なっている場合があります。
- ・ DOS/V 解説書としての体裁を整えるために、あえてシステムの位置付けを変更した部分があります。たとえばマウス制御はドライバで行うものですが、これは構成上「マウス BIOS」という項目に変えてあります。
- ・ サンプル・リストは実際に使用される方も多いため、それぞれ単独のソースとして記述しました。また、現実的なアセンブラでは DS(データ・セグメント)は CS(コード・セグメント)と一致させて記述することが多いため、DS はサンプル・リストではあえて設定しませんでした。
- ・ 本書内での数値の表現は基本的に 16 進数です。これはソフトウェアを作成される方にとっては、16 進数での記述が最も多いと考えてのことです。
- ・ 本書執筆中に発表された高解像度対応規格「V-Text」についても、詳細な解説を掲載しました。

読者の方々にはこれらの主旨をよくご理解いただき、本書を DOS/V アプリケーション開発の一助にいただければ、著者としてこれ以上の喜びはありません。

CONTENTS

第 1 章	DOS/V 概説	15
1.1	IBM-PC	16
	1.1.1 DOS/V 前夜	16
	1.1.2 初めに IBM-PC ありき	17
	1.1.3 PC 互換機の登場	19
1.2	DOS/V の発想	24
	1.2.1 日本語処理と高解像度	24
	1.2.2 舞台は日本へ	25
1.3	DOS/V 標準規格	28
1.4	DOS/V の基本原理	30
	(1) フォント・ドライバ(\$FONT.SYS)	
	(2) ディスプレイ・ドライバ(\$DISP.SYS)	
	(3) 入力支援ドライバ(\$IAS.SYS)	
	(4) プリンタ・ドライバ(\$PRNESC.P.SYS)	
第 2 章	プログラミング・ガイドライン	35
2.1	ソフトウェア割り込み	37
2.2	日本語 DBCS 処理	38
	(1) 2 バイト文字コードの問題	
	(2) 2 バイト文字の表示の問題	
	(3) 2 バイト文字の編集の問題	
	(4) パス名分解の問題	
2.3	文字コード体系	42
	(1) 制御コードの問題	
	(2) 特殊 1 バイト文字コード	
	(3) フォント字形の相違	
	(4) メーカー選定文字の相違	
	(5) ユーザ外字領域の相違	
	(6) JIS 制定年度による相違	

2.4	画面表示	44
	(1) ビデオ・モード (2) ビデオ・バッファ (3) カーソル表示 (4) システム予約領域 (5) エスケープ・シーケンス	
2.5	キーボード入力	48
2.6	プリンタ出力	55
	(1) 複数プリンタの管理 (2) コード変換 (3) 各社選定文字やユーザ定義文字の処理 (4) 非日本語プリンタへの対応 (5) プリンタ・ステータスの取得	
2.7	マウス	57
2.8	V-Text	58
	(1) 高品位テキスト・モードへの対応 (2) 高密度テキスト・モードへの対応	
2.9	Windows 対応	62
第 3 章	システム・コール	63
3.1	プログラムの終了(INT20H)	65
3.2	ファンクション・コール(INT21H)	66
3.3	終了アドレス(INT22H)	85
3.4	Ctrl-Break 割り込みアドレス(INT23H)	86
3.5	重大エラー・ハンドラ(INT24H)	87
3.6	絶対ディスクの読み書き(INT25H/INT26H)	92
3.7	プログラムの常駐終了(INT27H)	95

3.8	バックグラウンド処理(INT28H)[非公開]	96
3.9	高速 1 文字出力(INT29H)[非公開]	98
3.10	コマンドの起動(INT2EH)[非公開]	99
3.11	多重割り込み(INT2FH)	100
	(1) PRINT.EXE(多重番号 01H)	
	(2) アイドル・コール(多重番号 16H)	
	(3) HIMEM.SYS(多重番号 43H)	
	(4) DOSKEY.COM(多重番号 48H)	
	(5) ハイ・メモリ(多重番号 4AH)	
	(6) タスク・スイッチャー(多重番号 4BH)	
	(7) KEYB.COM(多重番号 ADH)	
	(8) APPEND.EXE(多重番号 B7H)	
第 4 章	BIOS コール	105
4.1	BIOS のワークエリア	108
4.2	ビデオ BIOS(INT10H)	112
4.3	装置構成情報(INT11H)	135
4.4	メモリサイズを得る(INT12H)	136
4.5	ディスク BIOS(INT13H)	137
4.6	RS-232C・BIOS(INT14H)	147
4.7	システム・サービス BIOS(INT15H)	150
4.8	キーボード BIOS(INT16H)	155
4.9	プリンタ BIOS(INT17H)	161
4.10	タイマ・クロック BIOS(INT1AH)	163

第 5 章	マウス BIOS	167
5.1	マウス BIOS の処理系	169
	<ul style="list-style-type: none"> (1) ビデオ・モード (2) カーソル (3) カーソル移動距離の単位 (4) マウスの動作確認 	
5.2	マウス BIOS の機能	172
第 6 章	V-Text	185
6.1	共通サブ・システムとのインタフェース	188
6.2	プリミティブ機能	193
6.3	ビデオ拡張プロファイル	200
第 7 章	メモリ・システム	203
7.1	EMS(Expanded Memory Specification)	205
	<ul style="list-style-type: none"> (1) ファイルのオープン機能を使った方法 (2) 割り込みベクトルを調べる方法 	
7.2	XMS(eXtended Memory Specification)	224
	<ul style="list-style-type: none"> (1) UMB(Upper Memory Block) (2) HMA(High Memory Area) (3) EMB(Extended Memory Block) 	
第 8 章	ハードウェア	233
8.1	割り込みコントローラ(i8259A)	236
	<ul style="list-style-type: none"> (1) イニシャライズ・コマンド・ワード (2) オペレーション・コマンド・ワード 	
8.2	DMA コントローラ(i8237A)	241

8.3	システム・タイマ(i8254)	243
	(1) モード・レジスタ(ライト) (2) ステータス(リード) (3) カウンタの読み書き(リード/ライト)	
8.4	リアルタイム・クロックと CMOS-RAM(MC146818)	246
	(1) CMOS-RAM のアドレス指定(ライト)	
8.5	システム・ポート	249
	(1) システム・ステータス(リード) (2) システム・コマンド(ライト)	
8.6	キーボード(i8042)	250
	(1) キーボード・データ(リード/ライト) (2) キーボード・コマンド(ライト) (3) キーボード・ステータス(リード)	
8.7	ビデオ(VGA)	253
	(1) テキスト・モード (2) グラフィック・モード (3) 汎用レジスタ (4) シーケンサー (5) ディスプレイ・コントローラ (6) グラフィック・コントローラ (7) 属性コントローラ (8) ビデオ DAC	
8.8	シリアル・ポート(NS16450)	281
	(1) デバイザラッチ・レジスタ(リード/ライト) (2) 通信割り込み制御レジスタ(ライト) (3) 通信割り込み表示レジスタ(リード) (4) 回線制御レジスタ(リード/ライト) (5) モデム制御レジスタ(リード/ライト) (6) 回線ステータス・レジスタ(リード/ライト) (7) モデム・ステータス・レジスタ(リード/ライト)	

8.9	パラレル・ポート(i8255A)	285
	(1) データ・レジスタ(リード/ライト)	
	(2) ステータス・レジスタ(リード)	
	(3) コントロール・レジスタ(リード/ライト)	
8.10	ディスク・コントローラ	289
Appendix		291
A- 1	文字コード表	292
A- 2	IBM 5576-A01 キーボード	302
A- 3	IBM U.S.English キーボード	306
A- 4	AX キーボード	310
A- 5	東芝 J-3100 キーボード	314
A- 6	DOS/V 非公式対応のキーボード	318
A- 7	世界各国のキーボード	321
A- 8	ANSI エスケープ・シーケンス一覧	335
A- 9	キーボード・コネクタ	340
A-10	ディスプレイ・コネクタ	341
A-11	シリアル・ポート・コネクタ	342
A-12	パラレル・ポート・コネクタ	344
A-13	ISA バスの信号位置	345
A-14	漢字コード表	347

●表タイトル一覧●

第1章

表 1-1	主要 IBM 純正機種のスเปック	21
表 1-2	OADG 参加会社一覧	28
表 1-3	OADG 規定の DOS/V ハードウェア・スเปック	29
表 1-4	\$FONT.SYS のパラメータ	31
表 1-5	\$DISP.SYS のパラメータ	32
表 1-6	\$IAS.SYS のパラメータ	33
表 1-7	\$PRNESC.P.SYS の パラメータ	33

第2章

表 2-1	CGA 文字モードとエミュ レート CGA 文字モード	45
表 2-2	キーコードの返らない 日本語入力キー	51
表 2-3	ワード入力による キーコード一覧表(提案)	53
表 2-4	日本語 FEP のモード変更操作	54
表 2-5	V-Text 対応の ビデオ・ボード一覧	58
表 2-6	V-Text 用に拡張追加 されたビデオ・モード	59
表 2-7	DSPX コマンドのパラメータ	59

第3章

表 3-1	DOS/V システム・コール一覧	64
表 3-2	ファンクション・コール(一般)	66
表 3-3	ファンクション・コール (IOCTL 関連)	78

表 3-4	ファンクション・コール (ネットワーク関連)	81
表 3-5	ファンクション・コールの エラー・コード	83
表 3-6	AX レジスタのエラー情報	87
表 3-7	DI レジスタのエラー情報	88
表 3-8	BP:SI レジスタで示される デバイス・ドライバ制御ブロック	88
表 3-9	AL レジスタの応答内容	88
表 3-10	新しい 32M バイト以上サポート のディスクへの読み書き	92
表 3-11	従来の 32M バイト以内の ディスクへの読み書き(参考)	94
表 3-12	常駐プロセスの インストール・ステータス	100
表 3-13	DOS/V で使用している 多重番号	100

第4章

表 4-1	IBM-PC の BIOS 一覧	106
表 4-2	BIOS 関連の割り込み	107
表 4-3	BIOS のワークエリア一覧	108
表 4-4	DOS/V のワークエリア一覧	111
表 4-5	ビデオ BIOS(INT10H) 機能一覧	112
表 4-6	DOS/V の日本語 ビデオ・モード	114
表 4-7	各ビデオ・モード時の 文字属性	114
表 4-8	パレットの初期値	119
表 4-9	カラー・レジスタの初期値	120

表 4-10	ディスク BIOS (INT13H) 機能一覧	137
表 4-11	ディスク BIOS の エラー・ステータス	138
表 4-12	フロッピーディスク・ドライブの パラメータ・テーブル	146
表 4-13	ハードディスク・ドライブの パラメータ・テーブル	146
表 4-14	RS-232C・BIOS (INT14H)機能一覧	147
表 4-15	RS-232C・BIOS の 回線制御状況ステータス	148
表 4-16	RS-232C・BIOS の モデム状況ステータス	148
表 4-17	システム・サービス BIOS の 機能一覧	150
表 4-18	キーボード BIOS (INT16H) 機能一覧	155
表 4-19	プリンタ BIOS (INT17H) 機能一覧	161
表 4-20	プリンタ BIOS の 状況ステータス	162
表 4-21	タイマ・クロック BIOS (INT1AH)機能一覧	163

第5章

表 5-1	各ビデオ・モードの解像度	169
表 5-2	マウス BIOS (INT33H) 機能一覧	172

第6章

表 6-1	V-Text により拡張・追加 されるビデオ BIOS	186
-------	--------------------------------	-----

表 6-2	拡張インタフェイス (INT15H) 機能一覧	188
表 6-3	プリミティブ機能一覧	193
表 6-4	ビデオ拡張プロファイルの エントリ一覧	200

第7章

表 7-1	DOS/V のメモリ・マップ	204
表 7-2	EMS の機能一覧	205
表 7-3	EMS のエラー・ステータス	223
表 7-4	XMS の機能一覧	224
表 7-5	XMS のエラー・ステータス	232

第8章

表 8-1	システム I/O アドレスの 使用状況の概要	234
表 8-2	割り込みレベル	236
表 8-3	DMA の利用状況	241
表 8-4	CMOS-RAM の内容	246
表 8-5	キーボードへのコマンド	251
表 8-6	キーボードからのコマンド	252
表 8-7	VGA テキスト・モード	254
表 8-8	カラーモードの アトリビュート	254
表 8-9	モノクロモードの アトリビュート	255
表 8-10	VGA の グラフィック・モード	255
表 8-11	各グラフィック・モード時の 基本色	255

●図タイトル一覧●

第 1 章		
図 1-1	DOS/V の デバイス・ドライバの概念	30
第 2 章		
図 2-1	推奨されるカーソル形状	46
図 2-2	システム予約領域の位置	46
図 2-3	IBM 5576-A01 キーボード	48
図 2-4	IBM U.S.English キーボード	48
図 2-5	AX キーボード	49
図 2-6	東芝 J-3100 キーボード	49
図 2-7	キーボード入力システムの概要	50
図 2-8	プリンタ出力のシステム	55
第 3 章		
図 3-1	ファイル属性	82
図 3-2	アクセス・モード	82
図 3-3	Ctrl-Break 割り込み処理	86
図 3-4	スタック上に保存される レジスタ値	89
第 4 章		
図 4-1	フォント・パターンデータの構造 (バイト・オフセット順)	152
第 5 章		
図 5-1	マウスの座標	169
第 6 章		
図 6-1	V-Text 対応ドライバの構造	186
第 7 章		
図 7-1	EMS の物理ページと 論理ページの関係	206
図 7-2	拡張メモリのイメージ	225
第 8 章		
図 8-1	キーボード・システム構成	250
図 8-2	VGA ビデオ・サブ・ システム	253
図 8-3	各グラフィック・モード時の VRAM 構造	256
図 8-4	書き込みモード 0 の基本動作	270
図 8-5	書き込みモード 1 の基本動作	272
図 8-6	書き込みモード 2 の基本動作	273
図 8-7	書き込みモード 3 の基本動作	274
図 8-8	プリンタ出力の タイミング・シーケンス	286

第 1 章

DOS/V 概説

1.1 IBM-PC

1.1.1 DOS/V 前夜

DOS/V の本当の姿を知ろうと思えば、そのすべての基礎となる IBM のパソコン「IBM-PC」を学ばないわけにはいきません。しかし、そう思い立って関連の技術資料を探してみると、その数のなんと少ないこと…。

残念なことに、この世界の標準機は、日本市場においてはこれまでほとんど気にも止められずにきました。いや正確には IBM の幾度にもわたるアプローチにも関わらず、日本のパソコン市場が頑として受け入れなかったというのが正しいのかもしれません。

IBM-PC およびその互換機は、欧米圏はもとより果ては共産圏にまで市場を拡大しておきながら、なぜか東洋のちっぽけな島国ニッポンだけには切り込むことができずにいました。その理由は3つありました。

第1の理由は、やはり日本が漢字という数千種にもおよぶ特殊な文字を使う国であったことです。さらに漢字は単に文字種が多いだけでなく、その構成が複雑なため1文字を表現するにはアルファベットの倍のドット数が必要です。これは1987年に発表されたVGAの登場まで、IBM-PCの世界では実現できない解像度でした。

第2の理由は、その日本語を容易に表示できる日本固有のパソコンが存在したこと（いうまでもなく NEC の PC-9801 等）。

パソコン上で自国の文字が利用できない国は何も日本だけではないのですが、だからといって独自のハードウェアをそう簡単に開発できるわけではありません。日本はそれが可能であったばかりに、独自の市場を形成することになってしまったのでしょう。

そして第3の理由は、日本人が他のどの国の人々よりも英語が不得意な国民だったことです。IBM-PC は英語圏のコンピュータです。ですからマニュアル類はすべて英語があたりまえなのですが、これが読めない。たとえば、

It was designed for children of all ages, and I hope they enjoy it.

など書くと、たいていの日本人はこれを読み飛ばしてしまいます（あなたはどうか？）。日本市場では日本語化は必須条件だったのです。

こんな状況下にある日本のパソコン市場でしたが、1980年代後半からのバブル景気をきっかけに多少状況が変わってきました。海外旅行ブームや英会話ブームによって多くの人が海外に出るようになり、その中のコンピュータに素養のある人々が、海外のパソコ

ン事情をみて不思議に思い始めたのです。

「世界中が IBM-PC を使っているのに、日本だけどうして違うの？」

「それに、どうしてこんなに安いんだろう？」

本当は日本のパソコンの価格が高すぎただけのことなのですが、日本の某メーカーの一派支配に飽き飽きしていた一部の人たちが、このパソコンを日本に持ち込みました。もちろん英語のみで使用するのですが、彼らは自分たちが海外で買ってきたソフトが、同じ日本語版よりもはるかに新バージョンであることにも気がつき始めました。

「なんで日本だけ発売が遅いんだろう？」

実はこんな状況にいちばん早くから気がつき、いちばん興味深く見守っていたのは、他ならぬ日本 IBM でした。IBM の一員でありながら、IBM-PC を商品として扱っていないのは唯一日本 IBM だけだったからです。

そのジレンマの中から DOS/V の発想が浮かび上がってくるのですが、それは次節にゆずるとして、まずはオリジナルの IBM-PC の歴史をみてみることから始めましょう。それがすべての始まりだったのですから。

1.1.2 ■ 初めに IBM-PC ありき

「IBM は、大胆にもパーソナルコンピュータ市場に進出を果たした。専門家たちはコンピュータの巨人 IBM が、この発展途上の若い業界でも 2 年以内に指導権を握るだろうと予測している。」

ウォール・ストリート・ジャーナル



写真 1-1

初代 IBM-PC、コードネームは「エイコーン」

1981年8月12日、ニューヨークのウォルドルフ・アストリア・ホテルで行われた、IBM初のパソコン「IBM-PC」の発表会を取材したある記者はこう書きました。

この予測は見事に的中することになるのですが、時代の主人公は決してIBM 1社だけではなく、多くの——これから新しく生まれる——互換機メーカーもその一翼を担うことになります。これは、これから始まる多くのアメリカンドリームへの幕開けでもあったのです。

このころのパソコンはまだマニアのおもちゃでしかなく、ビジネスなどの実用用途に使える代物ではありませんでした。すでにビジネス用計算機の世界で確固たる地位を築いていたIBMからみても、それは「取るに足らないもの」だったはずです。

そんな状況をビジ・カルクという表計算ソフトが変えてしまいました。このソフトの登場によりパソコンはビジネスの道具となり、パソコン市場の将来性は大きく変わり始めました。先進的なビジネスマンが、このソフトを利用するために競ってAppleのパソコンを購入し始めたことが、IBMのビジネス市場にも少なからぬ影響をおよぼし始めたのです。

そこで、IBMはパソコン市場への参入を決定しました。正直なところ、超優良企業IBMのビジネス市場を、ガレージでキットを作って売っている連中が脅かし始めたのですから、IBMとしては黙っているわけにはいかなかったのでしょう。

1980年8月、IBMは社内でさえも極秘のプロジェクト「チェス」をスタートさせました。リーダーのドン・エストリッジ以下、社内でも「変わり者」と呼ばれる12人のメンバに課せられた使命は、1年以内にビジネス市場に通用するIBM独自のパソコンを開発することでした。

コードネーム「エイコーン」と呼ばれたそのパソコンは、初めからオープン・アーキテクチャのシステムとして設計が行われました。これはつねに自社独自のハードウェアに固執するIBMとしては非常に珍しいことでしたが、このことが後にIBM-PCを業界標準^{*1}、しいては世界標準へと導くこととなります。もっともオープン・アーキテクチャにした本当の理由は、汎用部品を多用することで開発時間を少しでも節約するためだったのですが…。

彼らはハードウェアをオープンにすると同時に、システムの中核となるOSも外注することになりました。第1候補として、当時8ビットパソコンの標準OSとなりつつあった「CP/M」が検討されましたが、契約の場に開発者のゲーリー・キルドールが現れなかったため^{*2}、最終的には第2候補であったマイクロソフトに発注が行われました^{*3}。このとき契約

○*1：誰かが決めたわけではないのに、売れ筋などで決まってしまった標準規格を業界標準(デファクト・スタンダード)と呼ぶ。

○*2：業界では有名な話で、これには諸説があるが、キルドールがIBMとの会議をすっぽかしたのは本当のようである。このときキルドールがIBMと契約をしていたら、今日のマイクロソフトは存在しなかったかもしれないといわれている。

○*3：実際に開発を行ったのは、シアトル・コンピュータ・プロダクツのティム・パターソン。マイクロソフトは依頼者名を伏せたまま版権を買い取った。このころからマイクロソフトは商才に長けていたようである。

された OS は後に単独で MS-DOS として発表されますが、これによってマイクロソフトがどれだけの成功を納めたかは、ここであえて説明するまでもないことでしょう。

こうして丸1年の後、予定から1日も遅れることなく、エイコーンは IBM のパソコン「IBM-PC」として発表されることとなります。

しかし、この新しく発表されたパソコンはコンピュータの巨人 IBM が開発したにしてはあまりにも無難で、目新しさのかけらもありませんでした。

そこでそれを皮肉って、発表の数週間後に当時パソコン界の雄であった Apple は次のような広告をウォール・ストリート・ジャーナルに掲載しました。

「IBM さんようこそ。

35 年前にコンピュータ革命が始まって以来の最もエキサイティングで最も重要な業界へようこそ。アメリカの技術を世界に送り出すため、多大な努力を払っていただき私たちが堂々と渡り合ってくださいをお願いします。」

このときは誰もが IBM-PC を軽視し、「IBM なんて相手ではない」と思い込んでいました。しかし、時代は確実に標準化への道を歩み始めていたのです。周辺装置から果てはビデオボードまでもが選択や増設ができ、またそれらを開発するための技術情報が完全に公開された IBM-PC は、誰もが予想しなかったほど反響を呼びました。何よりもビジネスの IBM が作ったパソコン、という安心感が大きかったのかもしれません。

IBM-PC の売り上げは IBM の見積りよりも1桁多く、発表後2年間で50万台以上が普及することとなります。

この状況をみて、後に Apple の会長となったジョン・スカリーは語っています。

「あのような広告を出したことは、赤ずきんちゃんが自分で狼を引き入れたようなものだ。自信をもつこととうぬぼれることとは、微妙な違いがある。私たちは皆よい教訓を得たと思う。」

IBM-PC の発表から2年後の1983年3月、勢いにのった IBM は後継機「IBM-PC/XT」を発表。このマシンはハードディスクをサポートしていたため、他のメーカーのパソコンとの差は歴然となりました。しかし、この爆発的ともいえるパソコンブームのさなか、IBM の売り上げにもかげりがみえ始めていました。

1.1.3 ■ PC 互換機の登場

1983年1月、テキサス州ヒューストンで1台のパソコンが発表されました。

そのパソコンは IBM-PC 用のソフトウェアやハードウェアをまったく同一に利用することができましたが…IBM の製品ではありませんでした。これを作った会社は「Compaq (コンパック)」。

初の IBM-PC 互換機(PC クローン)が登場したのです。

IBM-PC はその仕様が公開されていた上に、使用した部品も簡単に入手できる汎用部品だったため、いとも簡単に複製できてしまったのです。

これを皮切りに、全米で有名無名の PC 互換機が製造・販売され、純正 IBM-PC と PC 互換機を合わせた台数は相当なものとなりました。その結果として、IBM-PC 用ソフトも多く開発されるようになり、市場は一気に賑やかになってきたのです。

1984 年 8 月、IBM はその後世界の標準機と呼ばれることになる「IBM-PC/AT」を発表しました。IBM-PC/AT は CPU に従来の 2 倍の処理能力をもつ i80286 を使用し、外部バスには初の 16 ビット規格の AT バスを採用し、後のボードの標準規格となります。

この IBM-PC/AT はそれなりの成功を納めますが、営業上は IBM がもくろんだほどのものとはなりません。

原因は PC 互換機でした。当時すでに PC 互換機は本家 IBM-PC よりも 30% ほど安く、かつ性能も互換性も問題がなくなり、おまけに Compaq などの一部のメーカー品は本家の性能をはるかに上回るものも販売を開始していました。これでは純正品が売れなくなるのは当然です。

そこで IBM は幾度となく、これらのメーカーに訴訟を起こしますが、もともと回路図は公開されたものであり、著作権訴訟上最大の論点となった「BIOS」も専門メーカーである Phenix や AMI など数社が勝訴したため、PC 互換機メーカーはこぞってこれらの BIOS メーカーから供給を受け、生産を続けることができました。

そんなある日の夕刻、フォートワース空港でデルタ航空 191 便が墜落しました。その乗客の中には、あの IBM-PC の生みの親ともいえるドン・エストリッジの名がありました。それは 1985 年 8 月、IBM-PC が発表されてからちょうど 4 年後のことでした。皮肉なことですが、これは 1 つの時代の終わりを意味していたのかもしれませんが。

彼は生前、あるインタビューに対し、IBM-PC の設計方針に関してこう語っています。

「他のものと違っているということが、犯してはならない最大の誤りだと私たちは確信していた。それはソフトウェアにしろハードウェアにしろ、IBM だけですべてを提供できるはずがないからだ。私たちは業界の標準を作りだそうなどとは考えていなかった。むしろ、すでに他社が確立したソフトウェアやハードウェアや販売網に合致するものをめざしたんだ。」



写真 1-2

IBM-PC の生みの親、ドン・エストリッジ

その後の 1987 年 4 月、IBM は再び自社シェアを奪回すべく、新しいバス・アーキテクチャである MCA(マイクロ・チャネル・アーキテクチャ)を登載した新型パソコン「PS/2」を発表しますが、このパソコンは互換機が簡単に作れないよう、そのアーキテクチャを非公開としていました。

この PS/2 はソフトウェア的には PC 互換機でもあり、MCA 自体も高性能なものであったので、それなりに評価は受けましたが、市場を独占しようとする IBM の態度はユーザの反感を買っただけで、IBM の評価は落ちる一方でした。

たしかに IBM-PC を開発したのは IBM でしたが、主導権はいつのまにか互換機メーカー……いえ、それを使うユーザに移っていたのです。

数年後、IBM は再び AT バス登載の PS/V を発表し、自らも互換機市場に参入せざるをえないはめになりました。本家が互換機を作る時代になってしまったのです。

表 1-1 主要 IBM 純正機種のスเปック

IBM-PC(Personal Computer)	
発表	1981 年 8 月
CPU	インテル 8088(4.77MHz)
主記憶	16K バイト(最大 512K バイト)
バス	8 ビット PC バス×5 スロット
補助記憶	カセットインタフェイス 5.25 インチ 1D(180K バイト) フロッピードライブ×1 基
ビデオボード	MDA, CGA
OS	PC-DOS1.1



写真 1-3

PC-XT

IBM-PC/XT(Personal Computer eXtended Technology)	
発表	1983 年 3 月
CPU	インテル 8088 (4.77MHz)
主記憶	128K バイト (最大 640K バイト)
バス	8 ビット PC バス×8 スロット
補助記憶	5.25 インチ 2D (360K バイト) フロッピードライブ×1 基 10M バイト ST-506 ハードディスク×1 基
ビデオボード	MDA, CGA
OS	PC-DOS2.1



写真 1-4

PC-AT

IBM-PC/AT(Personal Computer Advanced Technology)	
発表	1984 年 8 月
CPU	インテル 80286(6MHz)
主記憶	128K バイト(最大 16M バイト)
バス	8 ビット PC バス×2 スロット 16 ビット AT バス×6 スロット
補助記憶	5.25 インチ 2HD(1.2M バイト) フロッピードライブ×1 基 20M バイト ST-506 ハードディスク×1 基
ビデオボード	MDA, CGA, EGA
OS	PC-DOS3.1

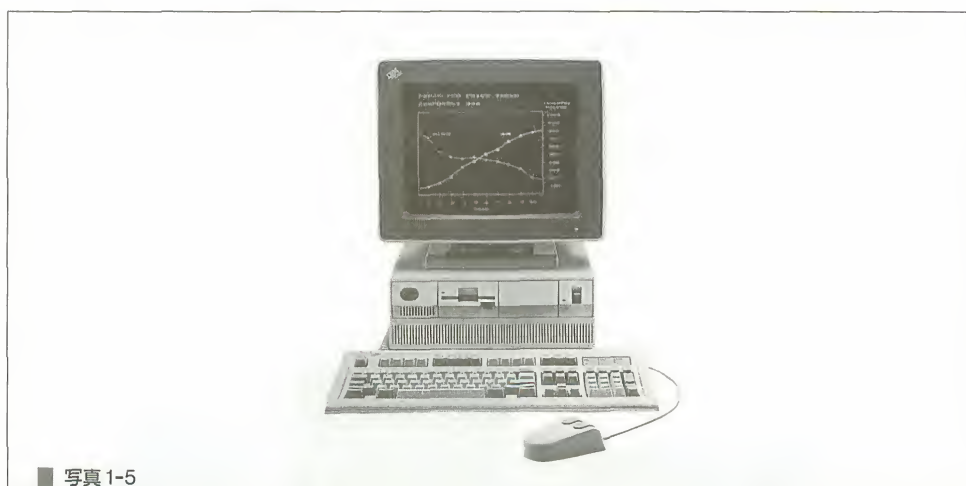


写真 1-5
PS/2

IBM-PS2(Personal Systems)	
発表	1987 年 4 月
CPU	インテル 80286(10MHz)
主記憶	1M バイト(最大 7M バイト)
バス	32 ビット MCA バス×3 スロット
補助記憶	3.5 インチ 2HD(1.44M バイト) フロッピードライブ×1 基 20M バイト ESDI ハードディスク×1 基
ビデオボード	VGA
OS	PC-DOS3.3, OS/2

1.2 DOS/V の発想

1.2.1 日本語処理と高解像度

1981 年、米国 IBM で IBM-PC の開発が行われていたころ、実は日本 IBM でもパソコンの開発がスタートしていました。しかし、そのパソコンはオープン思想を取った IBM-PC とはまったく異なり、完全に非公開のクローズド・アーキテクチャで設計されていたのです。

そのパソコンは、単にパソコンとしてだけでなく、IBM の汎用大型機の標準的端末 3270 の機能、ワープロとしての機能をもち、1 台で 3 役をこなすことから「マルチステーション 5550」と命名されました。つまりスタンドアロンではなく、あくまで大型機の端末として利用することが主目的のパソコンだったのです。もちろん、このころはまだ PC-DOS などありませんから、すべては独自のプログラムで起動し、利用する形態を取っていました。



写真 1-6

マルチステーション 5550

マルチステーション 5550 は当時の方式設計課長丸山力を設計責任者に、日本 IBM の藤沢研究所の精鋭たちを集めて開発が開始されました。このとき最大の論議の的となったのが画面解像度をどうするかという問題でした。

数々の論議の末、日本語を理想的に表現するには、横 40 文字、それに禁則処理用の 1 文字を追加した 41 文字×26 行。また文字フォントは日本語でいちばんよく使われている明朝体を表現するために 1 文字あたり 24×24 ドット。最終的には 1024×768 ドットもの

高解像度が必要ということになりました。

これは同時期に発表されていた IBM-PC 用の EGA の倍以上の解像度で、後に DOS/V の基準となる VGA の 640×480 ドットさえも上回っています。したがって、当時としては技術的にも困難な破格の高解像度だったのですが、「まともな日本語」が表示できるようにと開発スタッフは最後までこの解像度にこだわり続けました。

実は SuperVGA や XGA の解像度も同じ 1024×768 ドットなのですが、この値はこのマルチステーション 5550 のこだわりが、後に発展したものだったのです。

マルチステーション 5550 は、当時のパソコンが 16×16 ドットフォントを基準としていて中で 1 人 24×24 ドットフォントを採用したことで、ひときわ精彩を放ち、日本のビジネス市場ではそれなりの成功を納めました。

また、発表の数年後にはその高解像度を利用して、IBM-PC では不可能だった韓国語版や中国語版の開発が行われるなど、後の DOS/V の発想の原点を生み出し、パイリングル・パソコンの可能性の片りんを見い出すことにもなりました。

1.2.2 ■ 舞台は日本へ

マルチステーション 5550 の開発が進行しているころ、IBM は太平洋を囲む地域を APTO(アジア・パシフィック・テクニカル・オペレーション)と呼ぶ組織として統括し、当時工業先進国として地位を確立しつつあった IBM の日本支社に、この地域の開発・製造権限を与えていました。IBM の一支社にすぎなかった日本 IBM の役割が大きく変わり始めたのです。

1987 年 4 月、米国 IBM は「PS/2」を発表。その翌月、日本 IBM はマルチステーション 5550 の後継「PS/55 シリーズ」を発表します。

PS/55 はこの時点ではまだ IBM-PC 互換機ではなく、あくまで 5550 の後継機でしたが、注目すべき点は逆に米国版の PS/2 のほうが、オプションの XGA でマルチステーション 5550 と同じ解像度 1024×768 ドットを採用していたことです。

そして、もっと驚くべきことには、PS/2 の CPU ボードやハードディスクが日本から供給されていたことでした。実は PS/2 自身が、日米 IBM の共同開発製品だったのです。

当時、米国の IBM-PC の解像度は EGA で 640×350 ドットで、そろそろ次世代へ移行する必要があり、日本 IBM も低価格の普及版用として低解像度のパソコンを開発する必要に迫られていました。そこで日米が共同で進めたのが PS/2 の開発であり、その中で採用されたのが中解像度の EGA を拡張して 640×480 ドットにした VGA と、マルチステーション 5550 と同様の高解像度を採用した XGA だったのです。

「VGA の登場で、いよいよ IBM-PC でも日本語表示が実現できる。これでやっと世界中が同じ土俵に立つことができる。」

VGA が登場したとき、パーソナルコンピュータ事業本部長に抜てきされていた丸山力人は、5 年越しで計画した新日本語 DOS「DOS/V」の着想を副社長の三井信雄に打診しました。DOS/V は 16 ドット×16 ドットの文字フォントを VGA のグラフィックに表示することで、米国の PS/2 ひいては IBM-PC で日本語環境を実現しようというものでした。

これにより、世界中に普及している IBM-PC 上で日本語が利用可能になるだけでなく、同様の方法で他の言語圏でも利用の可能性ができ、IBM のパソコンは初めて世界的に統合されることになるのです。

しかし、これに難色を示す声も日本 IBM の社内には多く存在していました。

DOS/V への移行はマルチステーション 5550 のソフト・ハード資産を捨て去ってしまうことを意味していたからです。

「どうしていまさら低解像度に戻す必要があるんだ。」

「今までの資産をムダにしてしまうのか？」

反対意見はもっともでしたが、誰よりもそれを痛感していたのはマルチステーション 5550 を設計しておきながら、DOS/V を提唱しなければならなかった丸山力本人でした。また皮肉なことに、この時期に大手電機メーカー数社が結集して IBM-PC 互換機をベースに、EGA に日本語機能を付加したパソコンを AX と名づけて発表しましたが、これが見事に失敗してしまっていたのも大きなマイナス要因でした。

しかし、世界と同じパソコン環境を日本にも普及させることは、世界の孤児とまであだ名されていた日本 IBM にとって、なによりの悲願でもありました。

日本 IBM は多くの問題を抱えながらも DOS/V の推進に踏み切りました。そのうえ副社長の三井信雄は、かつて IBM-PC がそうしたように、DOS/V ではすべて情報を公開する方針を下しました。

「公開により日本 IBM は自分自身の市場を失うかもしれない。しかし DOS/V は日本市場のためだけではない。これは将来国際標準の DOS となりえる。アーキテクチャの標準化はメーカーの責任であり、日本 IBM は我が身を削ってでも共通化に踏み切った。今は市場を独占することよりも、拡大することに力を注ぐべきだ。」

1990 年 10 月、DOS/V は発表されました。

そして翌年の 1991 年 6 月、東京大手町の農協会館国際会議室で、DOS/V の推進団体である OADG(PC オープン・アーキテクチャ推進協議会)の結成記者会見が行われました。

OADG という中立機関を通じて DOS/V の仕様は、すべてのメーカーやユーザに公開されることとなったのです。

参加企業は日本 IBM を筆頭に、三菱電機、東芝、日立製作所、シャープ、松下電器産業、三洋電機、沖電気工業など、国内を代表するコンピュータや家電メーカーが結集しました。

それは、パソコン新時代を予見させるに十分な演出となりました。DOS/V は日本のパソコン業界に大きなインパクトを与えることとなり、日本電気を除くほとんどのパソコンメーカーが DOS/V 陣営を築くこととなりました。

DOS/V は、この日本語版を皮切りに、韓国語版や中国語版が発表され、現在はヨーロッパ語圏をターゲットに普及が見込まれており、文字どおり世界の標準 DOS としての地位を確保し始めたのです。

DOS/V について、日本 IBM のパソコン事業のボスである三井信雄副社長はこう語っています。

「DOS/V は、何も突然の思いつきで生まれたものではありません。IBM の本流は汎用大型計算機だと思われがちだが、日本 IBM の藤沢・大和の研究所の連中は、創立当初から『大型機はおれたちの仕事ではない。おれたちは、よその国ではできないことをやろう』と、これまでの研究を続けてきた。営業が大型機、大型機と騒いでいる間も、『あいつら何を儲けにならないことやってんだ』といわれつつも、20 年間も研究を積み重ねてきたんです。これは一朝一夕にはできない。今からではきっと米国の IBM 本社にもできないでしょう。それがやっとなんて大きく花開こうとしているだけです。」

1.3 DOS/V 標準規格

DOS/V は、世界の標準機と呼ばれる IBM-PC 上でも日本語が利用できるように考慮された DOS です。DOS/V は、もともと多国語サポートが可能のように設計されているため、これを世界共通のプラットフォームにすべく、その仕様の共通化団体として OADG が設立されました。

OADG = PC Open Architecture Developers' Group
(PC オープン・アーキテクチャ推進協議会)

表 1-2 OADG 参加会社一覧

インテルジャパン	AST リサーチ・ジャパン
沖電気工業	オムロン
キャノン	三洋電機
シャープ	ソニー
大同日本	デルコンピュータ
東芝	日本アイ・ビー・エム
日本エイサー	日本オリベッティ
日本デジタル・イクイップメント	日本ユニシス
日立製作所	ブラザー工業
マイタックジャパン	松下電器産業
三菱電機	メモレックス・テレックス
ユニシス・ジャパン・リミテッド	リコー
ワコム	

(平成 5 年 6 月 1 日現在 25 社 敬称・(株)省略 アイウエオ順)

OADG はソフトウェア利用の共通基盤の確立を主要な設立目的とし、「これにより異なるハードウェア上で稼働する多様なアプリケーションの提供が可能となり、パーソナルコンピュータの活用度が向上する」としています。

OADG は、DOS の「オープンな世界の確立のため」の世界唯一の標準化団体であるため、DOS/V 対応のソフトウェアを作成する際には、このガイドラインに従う必要があります。

逆にいえば、このガイドラインに従っている限り、OADG 承認のパソコン (OADG のロゴマークが入っている) での動作が保証されることになります。

OADG が規定している DOS/V が利用可能なハードウェア環境は次のとおりです。

表 1-3 OADG 規定の DOS/V ハードウェア・スペック

本体	IBM PC/AT・PS/2, およびその互換機
CPU	i80286 以上
バス	ISA バス (AT バス)
主記憶	640KB +256KB または漢字フォント ROM
キーボード	IBM 5576-A01 キーボード IBM U.S.English キーボード AX 仕様キーボード 東芝 J-3100 キーボード
プリンタ	エプソン ESC/P J84
文字コード	JIS-X0208 準拠のシフト JIS
メディア	1.44MB または 720KB フロッピー (1.2MB はオプション)
以上 1992 年 10 月仕様	

また、これらのハードウェア環境上で利用可能な DOS/V として、以下のものが発表されています。

DOS/V J5.0 (日本語版)

DOS/V T5.0 (台湾繁体字版)

DOS/V K5.0 (韓国語版)

DOS/V P5.0 (中国語版) (1993 年 4 月現在)

現在、ヨーロッパ圏向けの DOS/V も企画されており、今後 DOS/V は世界の標準 DOS としての地位を着実に固めていくでしょう。

OADG のガイドラインに従ったアプリケーションは、メッセージなどの問題を除けば、これらのすべての DOS/V で利用可能となります。

1.4 DOS/V の基本原理

「IBM-PC で日本語が使える」というふれこみで登場した DOS/V ですが、その拡張は、本来の IBM-PC の BIOS (基本入出力システム) に日本語対応のパッチを当てることで実現しています。

これには、新たに次の 4 つのデバイス・ドライバを常駐させることで対処しています。

フォント・ドライバ	: \$FONT.SYS
ディスプレイ・ドライバ	: \$DISP.SYS
入力支援ドライバ	: \$IAS.SYS
プリンタ・ドライバ	: \$PRNESC.P.SYS

これらのドライバが、相互にかつ複雑に関わることで日本語表示が実現されていますが、これらは特定の順序で常駐しなければ、機能が有効となりません。

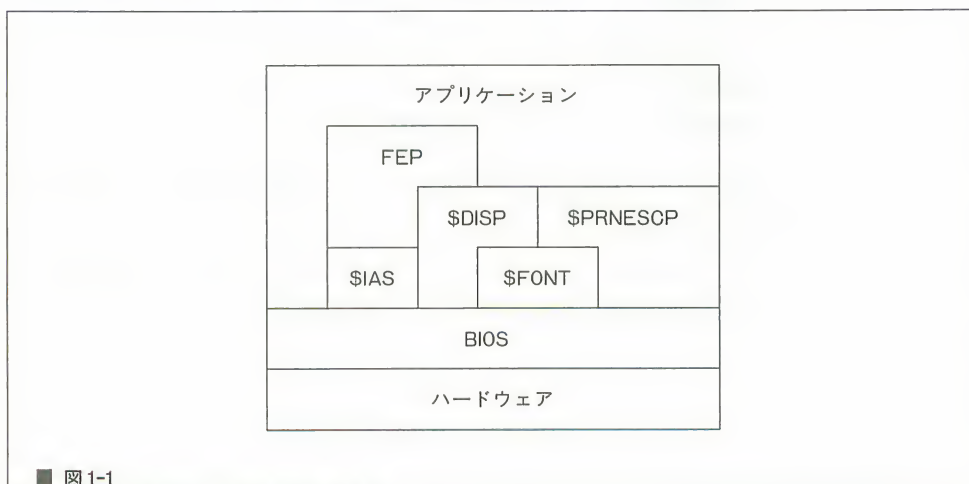


図 1-1

DOS/V のデバイス・ドライバの概念

(1) フォント・ドライバ(\$FONT.SYS)

日本語を表示するには、日本語の文字フォントが必要です。これをフォント・ファイルの形式でもち、起動時にメモリに読み込んで管理するのが、フォント・ドライバです。具体的には INT15H (システム・サービス入出力) を拡張しています。

フォント・ファイルは巨大なため、メモリ・アドレスの 1M バイトを越える領域に読み

込むことで、DOS のメイン・メモリを圧迫しないように配慮されています。

日本製のパソコンの場合は漢字 ROM が搭載されている場合も多いので、この場合はフォント・ファイルなしでも表示が可能となります。

画面表示用には通常 16×16 ドットのフォントを使いますが、非日本語対応プリンタを使用する際には、印刷用として 24×24 ドットのフォントも定義しておく必要があります。

このドライバはディスプレイ・ドライバやプリンタ・ドライバから呼び出されるので、すべてのドライバに先だって常駐させておく必要があります。

表 1-4 \$FONT.SYS のパラメータ

/24	[ON, OFF] 24×24 ドットフォントを使用するかどうか指定します。 デフォルトは OFF です。	
/F	[HN, ZN] 漢字フォント ROM より優先して、フォント・ファイルを使用するかどうか指定します。 HN は半角ファイルを、ZN は全角ファイルをロードすることを指定します。	(非公開)
/MSG	[OFF] 起動時のメッセージを非表示に指定します。	(非公開)
/U	? ユーザ・フォントの文字数を指定します。 デフォルトは 658 です。	
/P	パス名 フォント・ファイルをロードするディレクトリを指定します。 パス名の最後には必ず「¥」が必要です。	(非公開)

(2) ディスプレイ・ドライバ(\$DISP.SYS)

「VGA のグラフィック画面に文字を描画することで、日本語の利用を可能とする」という DOS/V の特徴を実現しているのが、このディスプレイ・ドライバです。具体的には INT10H (ディスプレイ入出力) を拡張しています。

ある意味では、この文字表示が DOS/V の動作速度を決定してしまうため、VGA のレジスタを直接コントロールしたり、属性比較を行ってできるだけ描画を少なくしたり、実際の表示プログラムではループを行わずに 1 直線型のプログラムが行われていたり、行スクロールに限ってハードウェア・スクロールを行ったりなどと、涙ぐましい方法で描画が行われています。

このため、一部の VGA 互換チップでは正常動作しない場合があり、そのための回避スイッチが設けられています。

また、VGA 上位互換のグラフィック・アクセラレータの場合はハード的にさらに高速化が可能なため、専用のディスプレイ・ドライバが付属している場合もあります。

表 1-5 \$DISP.SYS のパラメータ

/CHECK	(DOS/V 拡張キットのみ) 現在のハードウェアで、ディスプレイ・ドライバが動作可能かどうかを検査します。 結果は次の戻り値により判断します。 0 : 動作可能 1 : 動作不可能 2 : 機能検査ができない
/FC	[ON, OFF] (DOS/V 拡張キットのみ) XGA の場合、全角フォントをキャッシュするかどうか指定します。 これにより全角文字の表示は高速化されますが、ビデオ・モード設定時に時間がかかるようになります。 XGA の場合、この指定に関わらず、半角フォントはキャッシュされます。
/HS	[LC, OFF] (非公開) ハードウェア・スクロールの制御方法を指定します。 LC を指定すると、APA スタート・アドレス・レジスタとライン・コンペア・レジスタを使用して、ハードウェア・スクロールを行います。 もし、画面にゴミが出る場合は、ハードウェア・スクロールをあきらめて OFF を指定し、ソフトウェア・スクロールを行います。
/HS	[ON] (DOS/V 拡張キットのみ) APA スタート・アドレス・レジスタのみ使用し、ライン・コンペア・レジスタは使用せずにハードウェア・スクロールを行います。
/MODE	[モード番号] (DOS/V 拡張キットのみ) SuperVGA を使用している場合に、ビデオ・モード番号を指定します。SuperVGA ではこの番号を指定しないと、正常に表示が行われないことがあります。
/MSG	[OFF] (非公開) 起動時のメッセージを非表示に指定します。
/R	(DOS/V 拡張キットのみ) 常駐を解除します。
/TS	[バッファ・サイズ] (DOS/V 拡張キットのみ) テキスト・バッファのサイズを指定します。初期値は 13056 です。
/?	(DOS/V 拡張キットのみ) ヘルプ情報を指定します。

(3) 入力支援ドライバ(\$IAS.SYS)

英語版キーボードなどからも、日本語入力 FEP が利用可能なようにインタフェイスを提供するのが、この入力支援ドライバです。このドライバにより AX 仕様キーボードや J-3100 キーボードなども含めて、キーボードのハード的違いを吸収した日本語入力 FEP が開発可能となります。

日本語入力 FEP は、このドライバの機能を利用するので、\$IAS.SYS は必ず日本語入力 FEP よりも先に常駐していなければなりません。具体的には INT16H(キーボード入力)を拡張しています。

表 1-6 \$IAS.SYS のパラメータ

/G	[0, 1] 変換入力する位置を指定します。 0 はコマンドライン入力, 1 はエコー入力を意味し, デフォルトは 0 です。 このスイッチは EMS 常駐時のみ有効です。
/R	[0, 1] ローマ字変換モードを指定します。 0 はかな変換モード, 1 はローマ字変換モードで, デフォルトは 0 です。
/X	[0, 1] 常駐するメモリ領域を指定します。 0 はメイン・メモリ, 1 は EMS を意味します。デフォルトは 1 です。
/K	US 英語キーボードへ対応します。 英語キーボードでも変換キーなどの割り付けが仮想的に行われます。

(4) プリンタ・ドライバ(\$PRNESC.P.SYS)

DOS/V の標準プリンタ・コード体系は, エプソン提唱の「ESC/P J84」規格と規定されています。これに対応して, 日本語出力ができるようにするのが, このドライバです。具体的には INT05H (ハードコピー) と INT17H (プリンタ入出力) を拡張しています。

ESC/P では漢字コードは JIS コードなので, このドライバはシフト JIS コードを受け取って, JIS コードに変換してからプリンタに出力しています。コントロール・コードと JIS コード (ただし, シフト・イン, シフト・アウトが必要) はそのまま出力されています。

表 1-7 \$PRNESC.P.SYS のパラメータ

/R	[0, 1] JIS-X0208 でコード・ポイントの変更が行われた 26 組の漢字の, JIS コードへの対応変換を行います。 0 は無変更, 1 は変更で, デフォルトは 0 です。
/U	外字定義コード 外字定義コードを 16 進数で指定します。 デフォルトは 777EH です。

第 2 章

プログラミング ガイドライン

「DOS/V プログラミングの真髄は互換性にあり」といわれるほど、きちんと互換性を確保した DOS/V 対応プログラムの開発には多くのノウハウが必要です。これは他のパソコンと異なり、DOS/V は多くのメーカーの互換機を対象としないためです。

互換性を高めるには多くの方法がありますが、基本は OADG 提唱のガイドラインに添ってプログラミングを行うことです。しかし、この OADG のガイドラインは難解な部分やあいまいな部分が多く、おまけにそれが膨大な提供資料の中に散在しているため、通読して理解するのは非常に困難です。

そこで本章では、そのガイドラインにあたる部分をまとめて取り上げました。互換性の高い DOS/V のアプリケーションを作成するため、本章だけは通読することをお勧めします。

2.1 ソフトウェア割り込み

DOS/V の最も基本的なインタフェースは、ソフトウェア割り込みです。DOS/V のソフトウェア割り込みには以下のものがあります。

- ① ファンクション・コール (INT21H)
 - ② システム・コール (INT20H, 22H ~ 3FH)
 - ③ BIOS コール (INT10H, 16H, 17H)
-
- ④ BIOS コール (INT15H)
 - ⑤ BIOS コール (その他の INT11H ~ 1FH)

このうち、OADG の DOS/V で共通と規定されているのは③までです。互換性の高いアプリケーションを作成するためには、同様の機能がある場合は①, ②, ③の中で、できる限り高位の機能を使用するようにしてください。

④, ⑤は、PC/AT 互換機と PS/2 互換機で一部機能が異なっているため、一般的には使用すべきではありません。どうしても使用する必要がある場合は、複数の対応方法を用意しておくべきでしょう (詳しくは第 4 章「BIOS コール」を参照)。

ソフトウェア割り込みを取り込んだデバイス・ドライバ等を作成する場合は、必ず元の割り込みをチェインするなどして、自作のプログラム側にない機能をサポートするようにします。

これらのソフトウェア割り込みだけでは実現不可能なプログラムを作成したい場合は、あまり望ましい方法ではありませんが、直接ハードウェアを参照するしかありません。この場合は、すべての IBM-PC 互換機で共通のハードウェア機能かどうかを、事前に厳重に確認しておくべきでしょう (詳しくは第 8 章「ハードウェア」を参照)。

2.2 日本語 DBCS 処理

DOS/V に限らず、2 バイト系文字を扱わなければならない日本語処理は、非常に困難です。これはアプリケーション側のみの問題ではなく、DOS 自身を日本語化する際にも相当な障壁となっていました。

「DOS を日本語化するには、単にメッセージを日本語に書き換えれば済む」と考えるのは大きなまちがいで、実際には、DOS 本体部分を相当手直しする必要が生じます。

それを DOS/V では「コマンド 1 つで日本語・英語モードを行き来できるようにしよう」というわけですから、問題はさらに複雑になります。

多少話はそれますが、DOS の日本語化を行う際に問題となる点を以下にあげてみましょう。

(1) 2 バイト文字コードの問題

DOS は、もともと英語のみの使用を想定して設計されていたので、7 ビットのアスキー文字 128 種しか考慮されていませんでした。

しかし日本語は、JIS に規定される文字を表現するために最低でも 12 ビットが必要でしたので、マイクロソフトが中心となって、2 バイト系文字コードであるシフト JIS コードを体系化しました。

シフト JIS コードとは、本来の JIS コードを巧みなビット操作でシフトさせることで、第 1 バイト目が 81H ~ 9FH, E0H ~ FCH で始まるように設定したものです。

このシフト JIS コード体系により、日本語の文字コード処理は一応の解決をみましたが、日本のパソコンが 80H 以降のコードにグラフィック・キャラクタを割り振っているように、IBM-PC でもいろいろなキャラクタを割り当てて利用してしまっています。結局これらの問題から、日本語版ソフトと海外版ソフトとの混用は事実上不可能となっていました。

(2) 2 バイト文字の表示の問題

DOS のファンクション・コールを利用して、画面に単純に文字列を表示する場合でも、最終カラムに全角文字は表示できません。強制的に半角 1 文字分スキップさせ、表示を行う必要があります。

(3) 2 バイト文字の編集の問題

コマンドラインからメッセージを入力しているときに、バックスペースが押された場合、直前が全角文字ならば2カラム分を消去しなければなりません。

ところが困ったことに、シフト JIS コード体系では直前の文字が全角文字なのかどうかを判別するには、文字列の先頭から検索を行う以外に方法がありません。

たとえば「株 A」という文字列の場合、単純に直前2バイトを判別するとシフト JIS コード 9441H の「尿」と勘違いしてしまい、「株」の第2バイトのみを誤って消去してしまうのです。

文字列	株 (8A94H)	A (41H)	尿 (9441H)
データの並び	8AH 94H	41H	94H 41H

混同してしまう

(4) パス名分解の問題

DOS では、パスは「¥」(米国では「\」)で区切ることになっています。ですから、パス名をディレクトリ名とファイル名に分解するのは、単純に終端からいちばん初めに出現する「¥」を探すことで容易に実現できます。DOS の内部でもこの方法は多用されていました。

しかし、「¥」の文字コード 5CH はシフト JIS コードの第2バイト目にも頻発するため、日本語 DOS では問題が起きます。

たとえば「¥申請」というファイル名の場合、「申」の第2バイトのほうが「¥」より先にひっかかってしまいます。

文字列	¥ (5CH)	申 (905CH)	請 (90BFH)
データの並び	5CH	90H 5CH	90H BFH

先に認識されてしまう

上記の問題があったため、DOS の日本語化の際には相当な手直しが必要となり、ましてや英語モードとの共存などおおよもつかないことでした。

しかし、これを実現すべく、DOS/V では DBCS ベクタという概念を導入しました。

DBCS = Double Byte Code Set
(2 バイト文字体系)

これは2バイト系文字コードの第1バイト目の範囲を示す値で、日本語版 DOS/V では

```
81H, 9FH
E0H, FCH
00H, 00H      (00H, 00H はターミネータ)
```

となっています。これが英語モードの場合は

```
00H, 00H      (00H, 00H はターミネータ)
```

となり範囲が指定されていません。したがって、全コードが1バイト文字とみなされます。この値をソフト側が参照して判断を行うことで、2バイト系文字の正常な判断が可能となるのです。

この DBCS ベクタへのポインタはファンクション・コール (INT21H, AX=6300H) で取得できます。

DOS/V は日本語だけではなく、今後数か国語に対応する予定ですが、その場合も各国の文字体系の事情に関わらず、この DBCS ベクタを変更するだけで正常動作が望めるということになります。

ただし、2バイト系文字の第2文字目かどうかの判断は、これまでの日本語処理同様に文字列の先頭から調べるしかありません。

List 2-1

シフト JIS 第1文字目検査プログラム(これまでのプログラムスタイル)

```
; DL = 検査文字
; シフトJISでなければキャリーを返す
;
KANJI_CHECK:
        CMP     DL, 81H          ; 80H～9FH検査
        JB      KANJI_CHECK_E
        CMP     09FH, DL
        JNB     KANJI_CHECK_E

        CMP     DL, 0E0H        ; E0H～FCH検査
        JB      KANJI_CHECK_E
        CMP     0FCH, DL
        JNB     KANJI_CHECK_E

KANJI_CHECK_E:
        RET
```

List 2-2

DBCS 第 1 文字目検査プログラム(これからのプログラムスタイル)

```

; DL = 検査文字
; DBCSでなければキャリーを返す
;
DBCS_CHECK:
        PUSH    AX
        PUSH    BX
        PUSH    SI
        PUSH    DS

        MOV     AX, 6300H           ; DBCSの取得
        INT     21H

DBCS_CHECK_L:
        CLD
        LODSW
        OR      AX, AX             ; ターミネータ検査
        JZ      DBCS_CHECK_ER

        CMP     DL, AL             ; 範囲検査
        JB      DBCS_CHECK_E
        CMP     AH, DL
        JNB     DBCS_CHECK_E

        JMP     DBCS_CHECK_L

DBCS_CHECK_ER:
        STC

DBCS_CHECK_E:
        POP     DS
        POP     SI
        POP     BX
        POP     AX
        RET

```

2.3 文字コード体系

DOS/V での文字コード体系は

日本語モード時 : 1 バイト文字はコード・ページ 932(日本語図形文字セット)

2 バイト文字は JIS-X0208 のシフト JIS

英語モード時 : 1 バイト文字はコード・ページ 437(IBM-PC 図形文字セット)

を採用しています(詳しくは「Appendix」を参照)。

文字コードに関しては以下の点を考慮してください。

(1) 制御コードの問題

コード 00H ~ 1FH, 7FH は制御コードとして働きますが、一部にディスプレイ用の罫線などの特殊コードを定義しています。ただし、これらのコードのプリンタへの出力は保証されていません。

(2) 特殊 1 バイト文字コード

コード 80H, A0H, FDH, FEH, FFH は米国・欧州用の特殊文字コードとして予約されています。アプリケーション作成時には、これらのコードを独自の制御コードとして使用しないでください。

これらのコードは、米国などでは文字列中に含まれていることがあり、データ交換の際に思わぬ結果を招くおそれがあります。これらのコードは、他の 1 バイト文字と同様に扱うようにしてください。

(3) フォント字形の相違

OADG ではフォントの字形は規定していないため、メーカーにより異なります。グラフィック上でフォント操作を行うアプリケーションは、この点に十分注意する必要があります。

(4) メーカー選定文字の相違

2 バイト文字コードの一部にはメーカーが独自に定義したものが 있습니다。

IBM 選定文字 : 386 文字 (FA40H ~ FC4BH)

AX・エプソン選定文字 : 82 文字 (8740H ~ 879CH)

東芝選定文字 : 206 文字 (81ADH ~ 859EH)

これらの文字は、すべての DOS/V 上で利用可能とは限らないので、アプリケーションの表示には使用しないようにしてください。

(5) ユーザ外字領域の相違

ユーザ外字領域は F040H からです。この領域は、ユーザが定義した文字を DOS が管理するための領域です。終端などの位置は不確定なため、アプリケーションが独自の文字を定義してはなりません。

一般的には日本語 FEP 側に管理を任せるべき部分です。

(6) JIS 制定年度による相違

JIS-X0208 の 1983 年および 1990 年の改訂で、文字コードの追加や一部の入れ替えが行われています。これらの文字は、万一入れ替わっても構わないように考慮してください (詳しくは「Appendix」を参照)。

2.4 画面表示

VGA による日本語表示は、DOS/V のいちばんの特徴のある部分であると同時に、最も注意を要する部分でもあります。何とんでもグラフィック画面に文字を書くことで、英語テキスト・モードと同じ状況をシミュレートしようというのですから、問題が発生しても当然といえます。

したがって、問題をできる限り回避するために、アプリケーションは VGA のハードウェアを操作すべきでなく、DOS/V のファンクション・コールまたは BIOS コールなどの DOS/V で規定された方法で文字表示を実現すべきです。

(1) ビデオ・モード

DOS/V で規定されている、2 バイト文字表示が可能な画面モードには次のものがあります。

モード 11H : 2 色カラーグラフィック

モード 12H : 16 色カラーグラフィック (30 行モード)

モード 72H : 16 色カラーグラフィック (25 行モード)

モード 03H : 16 色エミュレート CGA テキスト・モード

モード 73H : 16 色エミュレート拡張 CGA テキスト・モード

このうち、DOS/V の標準文字モードは「モード 03H」です。これは IBM-PC 互換機の世界では最も一般的な文字モードで、DOS/V ではこれをグラフィック画面上でシミュレートする形で近似の機能を実現しています。

したがって、アプリケーションは、基本的にこのモードで設計されるべきです。しかし、確実な動作を保証するためには、開始時に現在の画面モードを取得して保存した後に、モード 03H を設定し、終了時には保存しておいた元の画面モードへ戻すようにしてください。

エミュレート CGA 文字モードは、実際の CGA テキスト・モードをあくまで近似的に実現したもので、表 2-1 のような相違があります。

表 2-1 CGA 文字モードとエミュレート CGA 文字モード

機 能	CGA 文字モード	エミュレート CGA 文字モード
ビデオ・バッファ ・セグメント	B800H	疑似ビデオ・バッファ (INT10H, AH=FEH で取得)
文字ジェネレータ	2 組	1 組
文字サイズ	8×8 ドット	8×19 ドット
文字の点滅属性	サポートあり	背景の輝度で代用
カーソルの点滅	前景色で点滅	点滅しない
ページ数	4 ページ	1 ページ

(2) ビデオ・バッファ

DOS/V では、文字モードをシミュレートするために、疑似ビデオ・バッファを用意しています。

疑似ビデオ・バッファのアドレスはビデオ BIOS の「ビデオ・バッファ・アドレスの読み取り」(INT10H, AH=FEH) で取得できます。これは、通常は英語版文字モードと同様の

B800H : 0000H

が設定されているはずですが、この値が返ってくるという保証はまったくないので、必ず確認すべきです。

この疑似ビデオ・バッファへ直接にコードや属性を設定した後、ビデオ BIOS「画面表示の更新」(INT10H, AH=FFH) で実際の描画を行うのが、最も高速に画面表示を行う方法ですが、この場合以下の問題を考慮する必要があります。

- ・ 2 バイト文字の第 1 バイト目と第 2 バイト目と同じ属性を設定する
- ・ 各行の先頭は必ず 1 バイト文字か、2 バイト文字の第 1 バイト目から始まるようにする
- ・ 文字列の先頭や終端で 2 バイト文字が重なった場合は、元の 2 バイト文字をスペースで完全に消去しておく

これらの問題を回避することと、英語モードとの互換性の観点から、文字表示は速度が特に問題にならない限り、すべての互換機で共通の機能であるビデオ BIOS の「文字列の書き込み」(INT10H, AH=13H) を利用することが推奨されています。

(3) カーソル表示

DOS/V では、カーソルはグラフィック画面に表示されている関係上から、ブリンクしません。

表示サイズも、BIOS の仕様上は任意の高さが設定できますが、実際は互換機のハードウェア上の制限から、以下の形状に制限することが推奨されています。これらの形状以外を指定した場合は、一部の互換機で動作が保証されない場合があります。デフォルトは底形です。

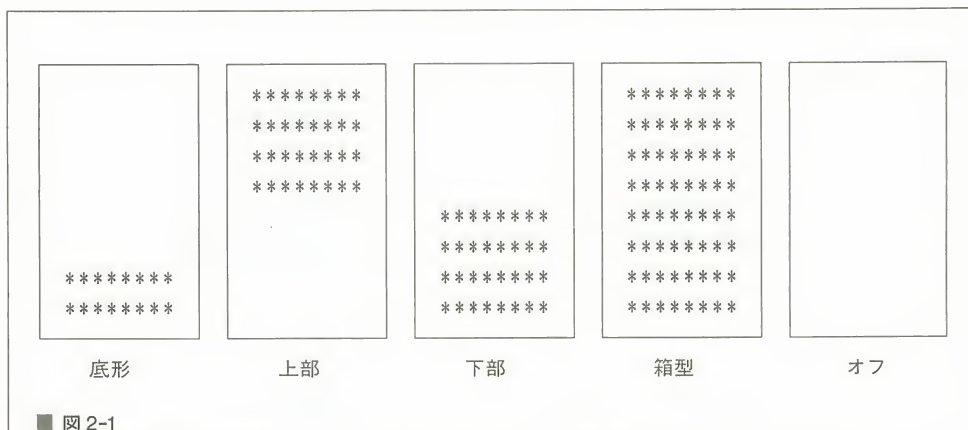


図 2-1

推奨されるカーソル形状

(4) システム予約領域

どのビデオ・モード時でも、再下行(25 行目, 30 行目等)はシステム領域として予約されており、日本語 FEP などが利用することになっています。

したがって、この領域はユーザは使用しないようにしてください。使用された場合の結果は予測できません。

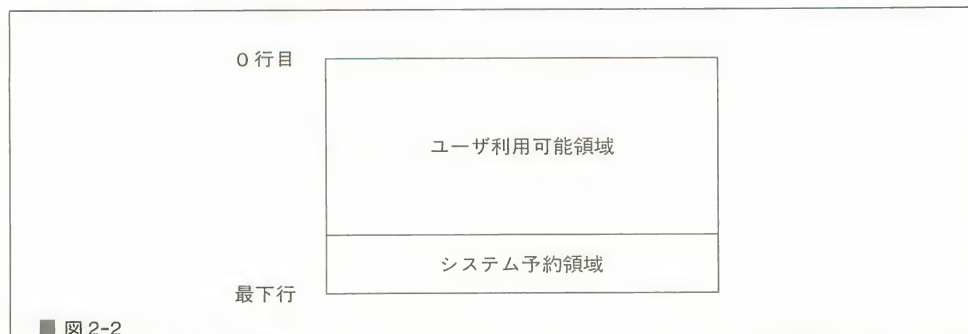


図 2-2

システム予約領域の位置

(5) エスケープ・シーケンス

エスケープ・シーケンスといえば、ANSI(米国国内規格協会)でも規定されている関係で、日本では「互換性のあるプログラム」を作成する方法の代表格のように受け取られていますが、あまり利用すべきではありません。

というのも、エスケープ・シーケンスは標準で用意されている機能ではなく、「ANSI.SYS」というドライバによって実現されている機能だからです。万一、このドライバが組み込まれていない場合は、画面にゴミが表示されることになります。

エスケープ・シーケンスで制御できる機能は、BIOSでも実現できるので、特殊な場合を除いては、あくまでBIOSを使用すべきです(詳しくは「Appendix」参照)。

2.5 キーボード入力

キーボード入力は対象となるハードウェアが数種類にもおよぶだけでなく、BIOS レベルでも扱うキーコードが異なるため、その対応には非常に慎重を要します。OADG では以下のキーボードをサポートすることとなっています。

- IBM 5576-A01 キーボード (106 キー)
- IBM U.S.English キーボード (101 キー)
- AX 仕様キーボード
- 東芝 J-3100 キーボード

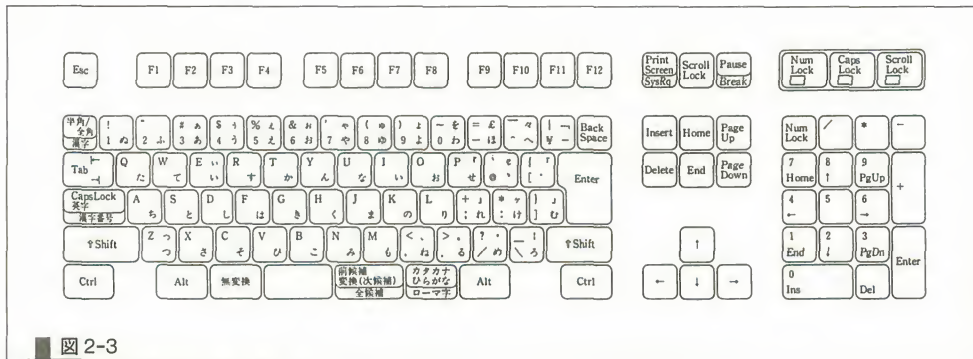


図 2-3
IBM 5576-A01 キーボード

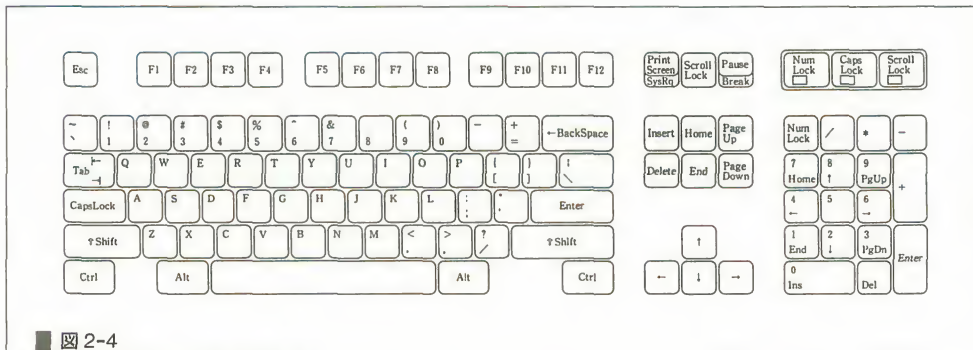


図 2-4
IBM U.S.English キーボード

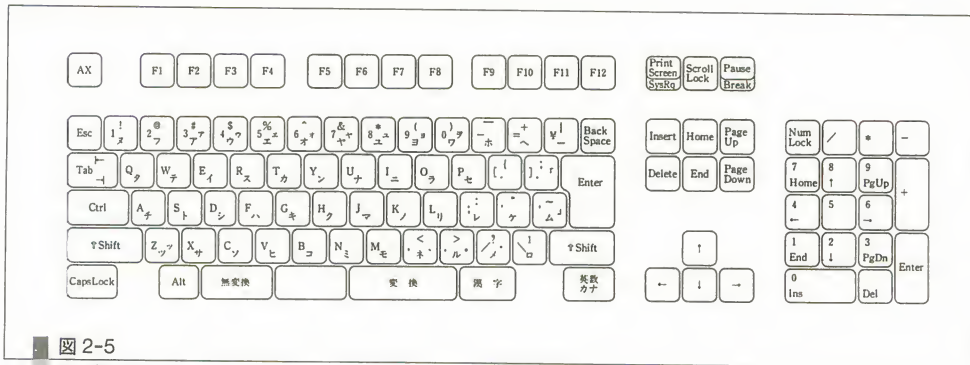


図 2-5

AX キーボード

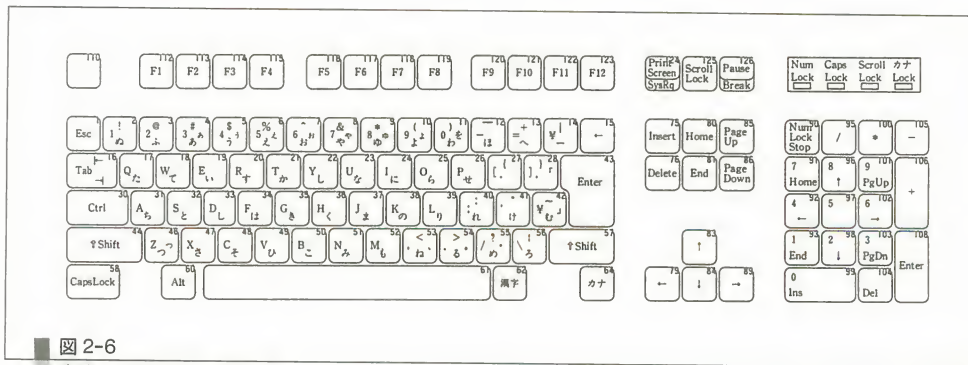


図 2-6

東芝 J-3100 キーボード

ところで、DOS/V は IBM が開発した関係上、非公式ですが以下のキーボードにも対応しています。しかし、OADG が標準としているのは、あくまで 5576-A01 キーボードなので、基本的にはこのキーボードを基準に対応を行うべきです。

- IBM 5576-001
- IBM 5576-002/003
- IBM 5553-S/5523

キーボードから返される走査コードは、それぞれのキーボードごとに異なります。これらの違いをできる限り吸収し、共通のインタフェースを取る方法が、DOS/V では確立されています。この処理は入力支援サブシステム(\$IAS.SYS)の常駐により提供されます。

キーボード入力は図 2-7 のように行われています。キーボードが押されると、ハードウェア割り込みとして INT09H が発生します。このとき取得されるデータは走査コード(押されたキーの番号、厳密にはメイク・コードが返ってくるが、説明を簡単にするために省略)で、その値はキーボードごとに異なります。この走査コードはコード・ジェネレータ

により文字コードに変換され、この時点で入力データは BIOS 管理下のキー・バッファに保留されます。

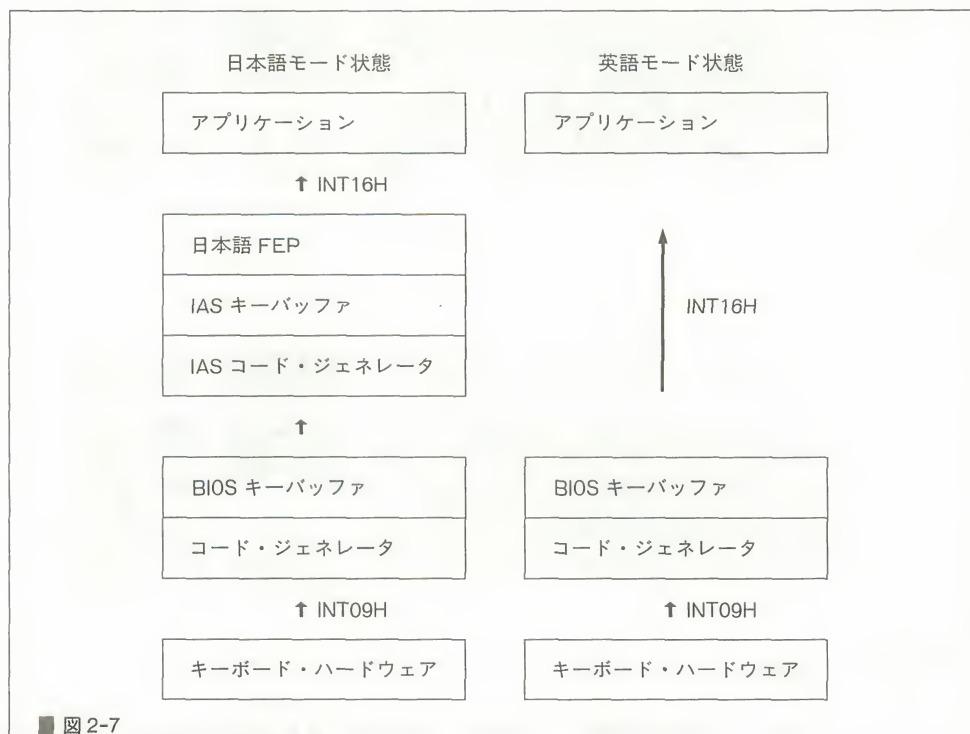


図 2-7

キーボード入力システムの概要

英語モード時は、ここでキーボード BIOS (INT16H) により入力データが読み取られるのを待ちますが、日本語モードの場合はここに入力支援ドライバ (\$IAS.SYS) が割り込んでおり、日本語 FEP などとの変換をした後のデータを、さらに入力支援ドライバ側のキー・バッファへ保留しています。

したがって、キーが押されていても、日本語 FEP による変換をユーザが確定していない間は、入力支援ドライバ側のキー・バッファには何も入っていない状態です。いくらキーボード BIOS を呼んでも何も入力がない状態が続くことになります。

また、日本語キーボードの場合は日本語変換に関わるキー (表 2-2 参照) のコードは返ってこなくなりますが、本来はこれらのキーは日本語 FEP 以外は利用すべきではないので、コードの返ってくる共通のキー範囲内でのみ、機能を割り当てるべきです。これは表 2-4 (p.54) に示す、入力支援システムにより提供される日本語 FEP の、モード変更操作キーについても同様です。

表 2-2 キーコードの返らない日本語入力キー

漢字	英数	カタカナ	ひらがな	カナ
英数カナ	変換	無変換	半角/全角	ローマ字
漢字番号	全候補	前候補	漢字制御	
Alt+'(IBM U. S. English キーボードの番号 1)				

これらの点を考慮した上で、互換性を考慮したキー入力を行うには、以下の手順で判断を行います。

仮に入力された走査コードを YY, 文字コードを XX とし、入力データを YY/XX と表現すると、次のようになります。

- ① 00/00 → Break
- ② 00/XX → 文字コード XX
- ③ YY/00 または YY/E0 → 拡張コード YY
- ④ YY/XX → 文字コード XX

このうえ、文字コードは DBCS の第 1 バイト目の可能性があるので、その場合は無条件にもう一度キー入力を行う必要があります。

どちらにせよ、互換性を維持するためには、アプリケーション側は走査コードではなく、文字コードで判断を行うプログラムを作成するべきです。

このようにしておけば、日本語 FEP にも完全対応ができ、さらにどの言語版の DOS/V にも対応が可能となります。

ここからは筆者独自の判断ですが、DBCS 対応を考慮するならば、最終的なキー取得データはすべて 16 ビットのワード扱いにするのが最良と思われます。

これは複雑なキー入力データの構成を単一化するためと、最近では大半のアプリケーションが C 言語で記述されていることから、その中で単純に符号なし整数 (unsigned int) で扱えるように配慮してのことです。

参考までに、DBCS 対応のワード単位キー入力プログラム (List 2-3) と、このプログラムを利用した場合のキーコード表 (表 2-3) を掲載します。

拡張コードは下位が 00H となるように設定しています。また、以下のコードは常識的に文字として入力できるので、表からは除外してあります。

0000H ~ 001FH : 制御コード

0020H ~ 00FFH : 文字コード (DBCS 第 1 バイト目も含む)

表 2-3 の空白の部分は、入力がないかあるいは入力判断すべきでない部分です。また * 印の付いたキーは、キーボードごとに配置が異なるので、たとえばダイヤモンド・カー

ソル機能用の配置位置が意味をもつ機能は割り当てるべきではありません。

IBM-PC では通常ファンクション・キーは F1 から F12 までありますが、F10 までの 10 キーしかない機種も存在するので、F11 や F12 は利用しないほうが望ましいと思われます。

また、Alt シフト+で返ってくるキーコードはキーボードごとに完全に異なるので、割り当てるべきではありません。

List 2-3

DBCS 対応ワード単位キー入力プログラム(提案)

```

; 返値  AX = キーコード
;
KEY_INPUT:
        PUSH    DX

        MOV     AH, 00H           ; キーボード入力
        INT     16H

        OR      AX, AX           ; Break 検査
        JZ      KEY_INPUT_E

        OR      AH, AH           ; 文字コード検査
        JZ      KEY_INPUT_F1

        OR      AL, AL           ; 拡張コード検査
        JZ      KEY_INPUT_E
        CMP     AL, 0E0H
        JNZ     KEY_INPUT_F1
        XOR     AL, AL
        JMP     KEY_INPUT_E

KEY_INPUT_F1:
        XOR     AH, AH
        MOV     DL, AL
        CALL    DBCS_CHECK      ; DBCS 検査
        JB      KEY_INPUT_E
        MOV     AH, 00H         ; 第2バイト目入力
        INT     16H
        MOV     AH, DL

KEY_INPUT_E:
        POP     DX
        RET

```

※ プログラム中のサブルーチン(DBCS_CHECK)はList2-2を参照

表 2-3 ワード入力によるキーコード一覧表(提案)

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
	3B00	3C00	3D00	3E00	3F00	4000	4100	4200	4300	4400
Shift	5400	5500	5600	5700	5800	5900	5A00	5B00	5C00	5D00
Ctrl	5E00	5F00	6000	6100	6200	6300	6400	6500	6600	6700
Alt	6800	6900	6A00	6B00	6C00	6D00	6E00	6F00	7000	7100

	1	2	3	4	5	6	7	8	9	0
		0300					001E			
Ctrl										
Alt	7800	7900	7A00	7B00	7C00	7D00	7E00	7F00	8000	8100

	Q	W	E	R	T	Y	U	I	O	P
	0011	0017	0005	0012	0014	0019	0015	0009	000F	0010
Ctrl										
Alt	1000	1100	1200	1300	1400	1500	1600	1700	1800	1900

	A	S	D	F	G	H	J	K	L
	0001	0013	0004	0006	0007	0008	000A	000B	000C
Ctrl									
Alt	1E00	1F00	2000	2100	2200	2300	2400	2500	2600

	Z	X	C	V	B	N	M
	001A	0018	0003	0016	0002	000E	000D
Ctrl							
Alt	2C00	2D00	2E00	2F00	3000	3100	3200

	[*	¥ *] *
Ctrl	001B	001C	001D
Alt			

	Esc *	Tab	Bs	INS	DEL	Home	End	PgUp	PgDn
Shift	001B	0009	0008	5200	5300	4700	4F00	4900	5100
Ctrl	001B	0F00	0008	5200	5300	4700	4F00	4900	5100
Alt	001B		007F			7700	7500	8400	7600

	↶	↑	↓	←	→	Pause	Print Screen
Shift	000D	4800	5000	4B00	4D00		
Ctrl	000D	4800	5000	4B00	4D00		
Alt	000A					0000	7200

表 2-4 日本語 FEP のモード変更操作

入力モード	IBM 5576-A01	IBM U.S.English	AX	東芝 J-3100
漢字	Alt + # 1	Alt + # 1	# 62	# 62
英数	# 30	Shift + # 30		
カナ	Shift + # 133	Alt + # 30		
ひらがな	# 133	Ctrl + # 30		
英数↔カナ ひらがな→カナ			# 64	# 64
英数↔ひらがな カナ→ひらがな			Shift + # 64	Shift + # 64
ローマ字	Alt + # 133	別途設定	Ctrl + # 64	Ctrl + # 64
半角↔全角	# 1	Ctrl + # 1	Shift + # 62	Shift + # 62
かな漢字制御	Ctrl + # 1	Alt Ctrl + # 1		

※ 別途設定は、メニューまたは CONFIG.SYS での設定を予定

※ #番号はキーコード番号。具体的には、それぞれのキーボードで以下のキートップのキーが該当する

キーボード	番号	仕 様
IBM 5576-A01	# 1	半角/全角(漢字)
	# 133	カタカナ・ひらがな(ローマ字)
IBM U.S.English	# 1	
	# 30	Caps
AX	# 62	漢字
	# 64	英数・カナ
東芝 J-3100	# 62	漢字
	# 64	カナ

2.6 プリント出力

IBM-PC では、プリントは実にさまざまな方式で接続が行われています。

- ・パラレル・ポート接続
- ・シリアル・ポート接続
- ・SCSI 接続
- ・その他

そこで DOS/V でもプリントのハード的な接続形態は規定しておらず、この違いはプリント・ドライバが吸収することになっています。

プリント・ドライバはプリント BIOS (INT17H) を拡張するので、プリント出力はすべてプリント BIOS を経由しなければ保証されないことになります。



図 2-8

プリント出力のシステム

プリントのコントロール体系は、エプソンの ESC/P J84 が標準に規定されており、以下のプリントがサポートされています。

- ・ESC/P J84 対応 日本語プリント
- ・ESC/P 対応 24 ドット 英語プリント

ところで、DOS/V は IBM が開発したものである関係上、以下のプリントもサポートしていますが、これは OADG では規定していないので、対応には注意が必要です。

- ・ IBM 5575/5577 系 日本語プリンタ
- ・ IBM PPS II Proprinter X24E/XL24E

プリンタ・ドライバ内では、以下の処理が行われます。

(1) 複数プリンタの管理

最大3台までのプリンタを、論理プリンタ・ポートという概念で管理しています。

(2) コード変換

ESC/P ではコード体系が JIS コードのため、シフト JIS コードからの変換を行います。JIS コードは変換の必要がないためパス・スルーでプリンタへ送ります。

(3) 各社選定文字やユーザ定義文字の処理

各社選定文字やユーザ定義文字を ESC/P の対応コードに変換するか、もしくは外字定義機能を使ってイメージ出力を行います。

この際、外字定義エリア (EC40H ~ EC9EH) をバッファとして使用します。

(4) 非日本語プリンタへの対応

英語版 ESC/P の場合は、フォント・ドライバの管理する 24 ドットフォントを利用してイメージ出力を行います。これにより非日本語対応プリンタでも日本語出力が可能となります。

(5) プリンタ・ステータスの取得

プリンタの動作状況を調査し、報告します。

プリンタ・ドライバでは、コントロール・コードや JIS コードは原則としてパス・スルーされますが、特殊なコントロールを行う場合は、コードによっては漢字イン・アウトの処理を行う必要がある場合があります。

2.7 マウス

IBM-PC では、マウスはプリンタと同様、どのような接続形態で実現されているか予測できません。したがって、すべての制御はマウス・ドライバ経由で行わなければなりません。

ただ、マウス・ドライバは、他のデバイス・ドライバ用の CONFIG.SYS に定義されるものではなく、実行形式で常駐する「MOUSE.COM」の形式を取っています。これは、マウスはすべてのアプリケーションでつねに必要とされるデバイスではないので、簡単に常駐・解放ができるようにするためです。

常駐 : MOUSE または MOUSE ON

解放 : MOUSE OFF

マウス・ドライバは、常駐することでマウス BIOS (INT33H) を拡張します。IBM の「PS/2」などでは独自に INT15H に BIOS を用意していますが、OADG のマウス BIOS は、あくまで INT33H に規定されているので、こちらを利用すべきです。仕様内容は、マイクロソフトのマウス・ドライバを基本にしています。

マウスは、アプリケーションがつねに利用するものではないので、ドライバ自体が組み込まれていない場合も多くあります。マウスを利用するアプリケーションは、安全のため、マウスが利用可能かどうかを以下の手順で判定すべきです。

① INT33H のベクタを検査する

もし 0000:0000H ならばマウス・ドライバが常駐していないので、処理を中断、もしくは警告を発する

② 初期設定を行う

アプリケーション開始時と終了時には、必ず「マウスの初期化 (INT33H, AH=00H)」を設定し、マウスが使用可能かどうかを判定する

マウスを利用するアプリケーションは、キーボードでも基本的な操作を行えるように配慮しておくべきでしょう。

2.8 V-Text

DOS/V の最大の魅力は、この「V-Text」かもしれません。V-Text は、もともとフリーウェアの作者たちが独自に企画した Hi-Text を、日本 IBM が正式に規格化したものです。これは DOS/V を拡張するキットの形式で提供されており、一部のドライバが拡張されています。また、これに伴い DOS シェルや CHEV コマンドも拡張されています。

この拡張により、XGA などの IBM 純正のビデオ・ボードはもとより、サード・パーティ製の SuperVGA や ET4000 等も正式にサポートされ、新たに以下の機能が追加されています。

[高品位テキスト・モード]

SuperVGA や XGA の高解像度モードを利用して、24 ドットの高品位な文字表示を行うモード。文字数自体は従来通りの 80 文字×25 行で、従来の DOS/V に準拠しているアプリケーションならば、特に変更もなく利用できる。

[高密度テキスト・モード]

SuperVGA や XGA の高解像度モードを利用して、より多くの文字の表示を行うモード。横 80 文字のままで縦方向のみを拡張した縦長モードと、縦横ともに拡張したワイド・モードがある。

表 2-5 V-Text 対応のビデオ・ボード一覧

ビデオ・ボード	高品位 テキスト	高密度テキスト	
		縦長	ワイド
XGA	80×25	80×38 80×42	128×42
XGA-2	80×25	80×33 80×38 80×39 80×42	100×33 106×39 128×42 160×56
VGA	未対応	80×34 80×40	80×34 80×40
SVGA	未対応	80×33 80×42 80×50	100×33 100×42 100×50
PS55 アダプター	80×25	80×38 80×42	128×42
ET4000	80×25	80×33 80×38	100×33 128×42

これらのモードの表示のために、次のビデオ・モードが拡張追加されています。

表 2-6 V-Text 用に拡張追加されたビデオ・モード

機能	モード	内 容
拡張	03H	高品位 (80×25) CGA 文字モード
	73H	高品位 (80×25) 拡張 CGA 文字モード
追加	70H	高密度 (可変サイズ) CGA 文字モード
	71H	高密度 (可変サイズ) 拡張 CGA 文字モード

これらのモードは、新たに追加された DOS/V の外部コマンドである DSPX で設定することができます。

表 2-7 DSPX コマンドのパラメータ

なし	メニュー・モードで設定を行います。
S	標準テキスト・モード (80 桁×25 行) に設定します。
L	縦長テキスト・モード (80 桁×多行) に設定します。
W	ワイド・テキスト・モード (多桁×多行) に設定します。
D	DOS/V 標準テキスト・モード (80 桁×25 行) に設定します (フォントは 16 ドット)。
/EXT	拡張テキスト・モードを指定します。上記のいずれのパラメータとでも組み合わせて指定が可能です。

V-Text は、これからの DOS/V の新しい流れとなることでしょうから、新規にアプリケーションを開発する際には、必ず対応すべきでしょう。

V-Text に対応するアプリケーションを作成する場合は、次の点を注意する必要があります。

(1) 高品位テキスト・モードへの対応

高品位テキスト・モードは、フォントが 24 ドットとなるだけで、画面表示は 80 桁×25 行と従来の DOS/V と同一です。したがって、DOS/V のガイドラインに添って作成されたアプリケーションであれば、問題なく動作します。

(2) 高密度テキスト・モードへの対応

高密度テキスト・モードでは、横桁数や縦行数が固定ではなく、使用するビデオ・ボードや各種設定により変化するので、これに対応した柔軟な設計が必要となります。

基本的に画面サイズの設定はユーザの選択に任せるべきなので、アプリケーション側は画面モードを操作すべきではありません (ユーザは DSPX コマンドで画面モード選択ができる)。アプリケーション側でどうしても特定の画面サイズを設定する必要がある場合は、その画面モードが利用可能かどうかを検査し、終了時には元の画面モードに戻すべきです。

高密度テキスト・モードに対応したアプリケーションを作成する際には、以下の点に注意する必要があります。

- ① 単純に文字列表示をする際に、横桁数が 80 文字を想定したような処理を行わない
(たとえば行末までスペースやタブでスキップすることなど)
- ② 起動時に横桁数と縦行数を取得し、以降はこの値を基準にして表示制御を行う

画面サイズ取得の方法 1：ワークエリアの参照

```
MOV    AX, 0040H
MOV    ES, AX
MOV    AL, ES:BYTE PTR [0049H]
MOV    BYTE PTR [ビデオ・モード], AL
MOV    AL, ES:BYTE PTR [004AH]
MOV    BYTE PTR [横桁数], AL
MOV    AL, ES:BYTE PTR [0084H]
INC    AL
MOV    BYTE PTR [縦行数], AL
```

画面サイズ取得の方法 2：BIOS コールによる取得

```
MOV    AH, 0FH
INT    10H
AND    AL, 7FH
MOV    BYTE PTR [ビデオ・モード], AL
MOV    BYTE PTR [横桁数], AH
MOV    AX, 1130H
MOV    BH, 01H
INT    10H
INC    DL
MOV    BYTE PTR [縦行数], DL
```

- ③ 起動時に画面モードの変更を行わず、画面の消去のみとしておく

画面の消去の方法

```
MOV    AX, 0600H
MOV    BH, 07H
XOR    CX, CX
MOV    DH, BYTE PTR [縦行数]
MOV    DL, BYTE PTR [横桁数]
SUB    DX, 0101H
INT    10H
```

- ④ アプリケーション上からコマンド・シェルを起動し、再びアプリケーションに復帰したときは、以前と異なったビデオ・モードに変更されている場合があるので、再度、ビデオ・モードや画面サイズの取得を行う

2.9 Windows 対応

ここ最近になって、日本でもやっと「Windows」が普及を始めたようですが、Windows を使い始めると、DOS/V のメリットはさらに大きくなります。何とんでも本家米国の IBM-PC とまったく同一のハードウェアを使っているのですから、あたりまえといえはそれまでですが…。

さて、Windows に対応したソフトウェアの開発ですが、ここでは本格的な Windows アプリケーションではなく、Windows の DOS シェル(DOS プロンプト)のウィンドウの中で、動作可能なアプリケーションを考えてみます。

Windows も Ver.3.1 より DOS シェル内で利用可能なフォント・サイズが可変となったため、ウィンドウサイズも従来の 640×480 ドット固定ではなくなりました。画面は相変わらず 80 桁×25 行ですが、Windows の特性からいって V-Text への対応も十分ありえるでしょう。

したがって、DOS シェルの中で動作可能なアプリケーションは、お行儀のよい(できれば V-Text 対応の)DOS/V アプリケーションであればよいといえます。

ただし、パレットの設定変更は実際には実行されませんし、キー操作も Alt+TAB 用のタスク切り替え用に予約されているものがあるので、注意が必要です。

第 3 章

システム・コール

DOS/V のシステム・コールは、俗にいう「MS-DOS」に相当する部分です。DOS/V は、あくまで IBM の PC-DOS をベースにしているので、システム・コールのみを DOS としてとらえる考えは取っていません。ですから DOS/V では、このシステム・コールを「DOS/V 割り込み」と表現しています。

DOS は BIOS より上位に位置し、大半の機能を BIOS の機能呼び出すことで実現しているため、より論理的かつ動的(ダイナミック)なシステムです。それゆえ OADG では、できる限り BIOS よりもこのシステム・コールを利用することを推奨しています。

特に、最近普及を始めた LAN(ローカル・エリア・ネットワーク)などへの対応については、NetWare を始めとするたいの NOS(ネットワーク OS)がこのシステム・コールを拡張しているので、それを考慮に入れると、BIOS 操作を行うことは多くの問題を生じさせる可能性があります。

DOS/V では、INT20H から INT3FH までをシステム・コールとして予約しています。これらは基本的に MS-DOS のシステム・コールそのものなので、あえて説明するまでもない内容なのですが、DOS/V になってからの改訂も少なからず発生しているため、確認の意味を含めて取り上げてみました。

なお、システム・コールに関する詳しい書籍はこれまで数多く刊行されているので、正確な詳細は他書に譲ります。本書では「すでに使うべきではないとされている機能」についての解説はあえて省略しますが、その対応策などを取り上げることで、現実的なプログラムのガイドラインとなるように考慮しました。

表 3-1 DOS/V システム・コール一覧

INT20H	プログラムの終了
INT21H	ファンクション・コール
INT22H	プログラムの終了アドレス
INT23H	Ctrl-Break の割り込みアドレス
INT24H	重大エラー・ハンドラ
INT25H	絶対ディスクの読み取り
INT26H	絶対ディスクの書き込み
INT27H	プログラムの常駐終了
INT28H	バックグラウンド処理(非公開)
INT29H	高速 1 文字出力(非公開)
INT2AH ~ 2DH	予約済み
INT2EH	コマンドの起動(非公開)
INT2FH	多重割り込み
INT30H ~ 3FH	予約済み

3.1 プログラムの終了 (INT20H)

このシステム・コールは使用すべきではありません。

この機能は CP/M との互換性を保つために残っているだけなので、DOS の Ver.2.00 以降は以下のファンクションを使用すべきです。

プロセスの終了 : INT21H, AH=4CH

常駐のままプロセス終了 : INT21H, AH=31H

3.2 ファンクション・コール(INT21H)

ファンクション・コールは DOS/V のサービス・ルーチンの中でも最も重要なもので、通常「DOS」といえばこのファンクション・コールを指します。

アプリケーションはこのファンクション・コールのみを使って記述されるのが理想ですが、現実的にはこれだけでは不可能なので、さらに低位の BIOS コールなどが使用されます。しかし、互換性のことを考慮すれば、同等の機能が実現されている場合は、できる限りこのファンクション・コールを使用すべきです。

ファンクション・コールに関しては、詳しく説明を行えばきりがなく、より詳しい書籍も多数出版されていますので、ここでは一覧のみにとどめておきます。

以下の一覧表の見方は、左端が機能番号、中央上部に機能名を記し、その下に入力パラメータとリターン情報とを併記しました。

なお、この一覧表では、新規に開発されるアプリケーションでは使用すべきでない機能(たとえば FCB 関連等)は、あえて削除してあります。

また、機能が OADG で非公開のものには「非公開」を、DOS/V あるいは DOS 5 特有のものに関しては「☆」印をつけました。参考にしてください。

機能 番号	機能名	非公開機能 → DOS/V あるいは DOS5 特有 → ☆
	入力パラメータ	リターン情報 ZF:ゼロ・フラグ CF:キャリー・フラグ

表 3-2 ファンクション・コール(一般)

01H	エコーつきキーボードの入力	
	AH=01H	AL=文字
02H	文字の出力	
	AH=02H DL=文字	なし
03H	補助入力	
	AH=03H	AL=文字
04H	補助出力	
	AH=04H DL=文字	なし

05H	プリンタへの出力	
	AH=05H DL = 文字	なし
06H	直接コンソール入力(入力待ち・エコー・Break なし)	
	AH=06H DL = FFH	ZF=0 の場合 AL=文字 ZF=1 の場合 入力文字なし
	直接コンソール出力(Break なし)	
	AH=06H DL = 文字(≠FFH)	なし
07H	直接コンソール入力(エコー・Break なし)	
	AH=07H	AL=文字
08H	キーボード入力(エコーなし)	
	AH=08H	AL=文字
09H	文字列の出力	
	AH=09H DS : DX = 文字列	なし
0AH	バッファつきキーボード入力	
	AH=0AH DS : DX = 入力バッファ	なし
0BH	キーボード・ステータスの検査	
	AH=0BH	AL=FFH : 入力文字あり ≠FFH : 未定義
0CH	バッファを空にしてのキーボード入力	
	AH=0CH AL = 機能番号 01H, 06H, 07H 08H, 0AH	不定
0DH	ディスクのリセット	
	AH=0DH	なし
0EH	デフォルト・ドライブの設定	
	AH=0EH DL = ドライブ番号 0 : A, 1 : B...	AL=論理ドライブの台数
19H	デフォルト・ドライブの取得	
	AH=19H	AL=ドライブ番号 0 : A, 1 : B...

1AH	ディスク転送アドレスの設定	
	AH=1AH DS:DX=ディスク転送アドレス	なし
1BH	デフォルト・ドライブ情報の取得	
	AH=1BH	AL≠FFH の場合 AL=セクタ数/クラスタ CX=バイト数/セクタ DX=クラスタ数/1ドライブ AL=FFH の場合 エラー
1CH	ドライブ情報の取得	
	AH=1CH DL=ドライブ番号 0:カレント, 1:A...	AL≠FFH の場合 AL=セクタ数/クラスタ CX=バイト数/セクタ DX=クラスタ数/1ドライブ AL=FFH の場合 エラー
1FH	デフォルト DPB アドレスの取得 ▶ 非公開 ◀ ☆	
	AH=1FH	AL=00H の場合 DS:BX=DPB アドレス AL=FFH の場合 エラー
25H	割り込みベクタの設定	
	AH=25H AL=割り込みベクタ番号 DS:DX=割り込みアドレス	なし
2AH	日付の取得	
	AH=2AH	CX=年(1980年基準) DH=月(1~12) DL=日(1~31) AL=曜日(0:日, 1:月...)
2BH	日付の設定	
	AH=2BH CX=年(1980年基準) DH=月(1~12) DL=日(1~31)	AL=00H:有効な日付 FFH:無効な日付

2CH	時刻の取得	
	AH=2CH	CH=時(0～23) CL=分(0～59) DH=秒(0～59) DL=1/100 秒(0～99)
2DH	時刻の設定	
	AH=2DH CH=時(0～23) CL=分(0～59) DH=秒(0～59) DL=1/100 秒(0～99)	AL=00H : 有効な日付 FFH : 無効な日付
2EH	ベリファイ・フラグの設定	
	AH=2EH AL=0 : ベリファイを行わない 1 : ベリファイを行う DL=00H	なし
2FH	ディスク転送アドレスの取得	
	AH=2FH	ES : BX=ディスク転送アドレス
30H	バージョン番号の取得 	
	AH=30H	AL=バージョンの整数部 AH=バージョンの小数部 BH=0 : RAM 上動作 1 : ROM 上動作 CX=0000H
31H	プロセスの常駐終了	
	AH=31H AL=リターン・コード DX=常駐するパラグラフ・サイズ	なし
32H	DPB アドレスの取得 ▶ 非公開 ◀ 	
	AH=32H DL=ドライブ番号 0 : カレント, 1 : A...	AL=00H の場合 DS : BX=DPB アドレス AL=FFH の場合 エラー

33H	Ctrl+Break の取得	
	AH=33H AL=00H	DL=ステータス 0:OFF 1:ON
	Ctrl+Break の設定	
	AH=33H AL=01H DL=0:OFF 1:ON	
	ブート・ドライブの取得 ☆	
	AH=33H AL=05H	DL=ドライブ番号 1:A, 2:B...
	真のバージョン番号取得 ☆	
	AH=33H AL=06H	BL=バージョンの整数部 BH=バージョンの小数部 DL=レビジョン・レベル DH=DOS フラグ ビット4:HMA で動作中 ビット3:ROM で動作中
34H	InDOS フラグ・アドレスの取得 ☆	
	AH=34H	ES:BX=フラグ・アドレス
35H	割り込みベクタの取得	
	AH=35H AL=割り込みベクタ番号	ES:BX=割り込みアドレス
36H	ディスクの空き容量の取得	
	AH=36H DL=ドライブ番号 0:カレント, 1:A...	AX≠FFFFH の場合 BX=使用可能なクラスタ数 DX=クラスタ数/1 ドライブ CX=バイト数/1 セクタ AX=セクタ数/1 クラスタ AX=FFFFH の場合 ドライブ番号が無効
38H	国別情報の取得	
	AH=38H AL=00H DS:DX=情報バッファ・アドレス	CF=0 の場合 BX=カントリ・コード CF=1 の場合 AX=エラー・コード

39H	ディレクトリの作成	
	AH=39H DS:DX=パス名のアドレス	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
3AH	ディレクトリの削除	
	AH=3AH DS:DX=パス名のアドレス	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
3BH	カレント・ディレクトリの移動	
	AH=3BH DS:DX=パス名のアドレス	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
3CH	新規ファイルのオープン	
	AH=3CH DS:DX=パス名のアドレス CX=ファイル属性	CF=0 の場合 AX=ファイル・ハンドル CF=1 の場合 AX=エラー・コード
3DH	既存ファイルのオープン	
	AH=3DH DS:DX=パス名のアドレス AL=アクセス・モード	CF=0 の場合 AX=ファイル・ハンドル CF=1 の場合 AX=エラー・コード
3EH	ファイルのクローズ	
	AH=3EH BX=ファイル・ハンドル	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
3FH	ファイルからの読み出し	
	AH=3FH BX=ファイル・ハンドル CX=読み込むバイト数 DS:DX=バッファ・アドレス	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード

40H	ファイルへの書き込み	
	AH=40H BX=ファイル・ハンドル CX=書き込むバイト数 DS:DX=バッファ・アドレス	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
41H	ファイルの削除	
	AH=41H DS:DX=パス名のアドレス	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
42H	ファイル・ポインタの移動	
	AH=42H BX=ファイル・ハンドル AL=0: ファイルの先頭から 1: 現在位置から 2: ファイルの終端から CX:DX=移動するバイト数	CF=0 の場合 DX:AX=ファイル・ポインタ CF=1 の場合 AX=エラー・コード
43H	ファイル属性の取得	
	AH=43H AL=00H	CF=0 の場合 CX=ファイル属性 CF=1 の場合 AX=エラー・コード
	ファイル属性の設定	
	AH=43H AL=01H CX=ファイル属性	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
44H	装置の IOCTL(詳細は表 3-3 参照)	
45H	ファイル・ハンドルの二重化	
	AH=45H BX=ファイル・ハンドル	CF=0 の場合 AX=新規のファイル・ハンドル CF=1 の場合 AX=エラー・コード
46H	ファイル・ハンドルの強制二重化	
	AH=46H BX=ファイル・ハンドル CX=新規のファイル・ハンドル	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード

47H	カレント・ディレクトリの取得	
	AH=47H DS:SI=バッファ DL=ドライブ番号 0:カレント, 1:A...	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
48H	メモリ・ブロックの割り当て	
	AH=48H BX=取得するパラグラフ・サイズ	CF=0 の場合 AX=ブロック・セグメント CF=1 の場合 AX=エラー・コード BX=割り当て可能な最大サイズ
49H	メモリ・ブロックの開放	
	AH=49H ES=開放するブロックのセグメント	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
4AH	メモリ・ブロックのサイズ変更	
	AH=4AH ES=ブロック・セグメント BX=変更するパラグラフ・サイズ	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード BX=割り当て可能な最大サイズ
4BH	プログラムのロードと実行	
	AH=4BH AL=00H DS:DX=パス名 ES:BX=パラメータ・ブロック	CF=0 の場合 ES:BX=PSP アドレス CF=1 の場合 AX=エラー・コード
	プログラムのロードと非実行) 非公開 (
	AH=4BH AL=01H DS:DX=パス名 ES:BX=パラメータ・ブロック	CF=0 の場合 ES:BX=PSP アドレス CF=1 の場合 AX=エラー・コード
	オーバーレイのロード	
	AH=4BH AL=03H DS:DX=パス名 ES:BX=パラメータ・ブロック	CF=0 の場合 ES:BX=PSP アドレス CF=1 の場合 AX=エラー・コード

	実行ステータスの設定		〉 非公開 〈
	AH=4BH AL=05H DS:DX=実行ステータス構造体	なし	
4CH	プロセスの終了		
	AH=4CH AL=リターン・コード	なし	
4DH	リターン・コードの取得		
	AH=4DH	AH=終了コード AL=リターン・コード	
4EH	最初に一致するファイル名の検索		
	AH=4EH DS:DX=パス名 CX=ファイル属性	CF=0 の場合 DTA=検索結果 CF=1 の場合 AX=エラー・コード	
4FH	次に一致するファイル名の検索		
	AH=4FH	CF=0 の場合 DTA=検索結果 CF=1 の場合 AX=エラー・コード	
50H	PSP アドレスの設定		〉 非公開 〈
	AH=50H BX=PSP セグメント	なし	
51H	PSP アドレスの取得		〉 非公開 〈
	AH=51H	BX=PSP セグメント	
52H	内部変数領域のアドレス取得		〉 非公開 〈
	AH=52H	ES:BX=内部変数アドレス	
53H	BPB から DPB への変換		〉 非公開 〈
	AH=53H DS:SI=BPB アドレス ES:BP=DPB アドレス	なし	
54H	ベリファイ状態の取得		
	AH=54H	AL=0:ベリファイ OFF 1:ベリファイ ON	
55H	PSP の複写		〉 非公開 〈
	AH=55H DX=PSP の複写先セグメント SI=DX:[0002H] の値	なし	

56H	ファイル名の変更	
	AH=56H DS:DX=現在のファイル名 ES:DI =新規ファイル名	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
57H	ファイル日付の取得	
	AH=57H AL=00H BX=ファイル・ハンドル	CF=0 の場合 CX=時刻 DX=日付
	ファイル日付の設定	
	AH=57H AL=01H BX=ファイル・ハンドル CX=時刻 DX=日付	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
58H	メモリのアロケーション方法の取得	
	AH=58H AL=00H	CF=0 の場合 AX=ビット 7 : UMB から ビット 6 : UMB だけ ビット 0 ~ 1 : 2=上位から 1=最小から 0=下位から CF=1 の場合 AX=エラー・コード
	メモリのアロケーション方法の設定	
	AH=58H AL=01H BX=ビット 7 : UMB から ビット 6 : UMB だけ ビット 0 ~ 1 : 2=上位から 1=最小から 0=下位から	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
	UMB のリンク・ステータスの取得	
	AH=58H AL=02H	CF=0 の場合 AL=リンク・ステータス CF=1 の場合 AX=エラー・コード

UMB のリンク・ステータスの設定		▶ 非公開 ◀
	AH=58H AL=03H BX=0: リンクしない 1: リンクする	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
59H	拡張エラー・コードの取得	
	AH=59H BX=00H	AX=拡張エラー・コード BH=エラー・クラス BL=推奨される訂正措置 CH=エラー発生箇所 (レジスタ破壊あり)
5AH	一時ファイルの作成	
	AH=5AH CX=ファイル属性 DS: DX=一時パス名バッファ	CF=0 の場合 AX=ファイル・ハンドル CF=1 の場合 AX=エラー・コード
5BH	新しいファイルの作成	
	AH=5BH CX=ファイル属性 DS: DX=パス名	CF=0 の場合 AX=ファイル・ハンドル CF=1 の場合 AX=エラー・コード
5CH	ファイル・アクセスのロック	
	AH=5CH AL=00H BX=ファイル・ハンドル CX: DX=ロック領域の位置 SI: DI=ロック領域のサイズ	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
	ファイル・アクセスのロック解除	
	AH=5CH AL=01H BX=ファイル・ハンドル CX: DX=ロック領域の位置 SI: DI=ロック領域のサイズ	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
5EH	ネットワーク・ファンクション(詳細は表 3-4 参照)	

60H	パス名の置換 ▶ 非公開 ◀	
	AH=60H DS:SI=パス名 ES:DI=置換後のパス名	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
62H	PSP アドレスの取得	
	AH=62H	CF=0 の場合 BX=PSP セグメント CF=1 の場合 AX=エラー・コード
63H	DBCS ベクタ情報の取得	
	AH=63H AL=00H	CF=0 の場合 DS:SI=ベクタ・テーブル・アドレス CF=1 の場合 AX=エラー・コード
67H	最大ハンドル数の設定	
	AH=67H BX=ハンドル数	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
68H	ファイル・バッファのフラッシュ	
	AH=68H BX=ファイル・ハンドル	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
6CH	ファイルの拡張オープン	
	AH=6CH AL=00H BX=オープン・モード CX=ファイル属性 DX=アクション DS:SI=パス名	CF=0 の場合 AX=ファイル・ハンドル CX=1: ファイルをオープンした 2: ファイルが作成された 3: ファイルが置換された CF=1 の場合 AX=エラー・コード

表 3-3 ファンクション・コール(IOCTL 関連)

44H	装置情報の取得	
	AH=44H AL=00H BX=ファイル・ハンドル	CF=0 の場合 DX=デバイス・データ CF=1 の場合 AX=エラー・コード
	装置情報の設定	
	AH=44H AL=01H BX=ファイル・ハンドル DX=デバイス・データ	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
	キャラクタ・デバイスからの制御文字列の読み取り	
	AH=44H AL=02H BX=ファイル・ハンドル CX=制御文字列のバイト数 DS:DX=制御文字列のアドレス	CF=0 の場合 AX=読み取ったバイト数 CF=1 の場合 AX=エラー・コード
	キャラクタ・デバイスへの制御文字列の書き込み	
	AH=44H AL=03H BX=ファイル・ハンドル CX=制御文字列のバイト数 DS:DX=制御文字列のアドレス	CF=0 の場合 AX=書き込んだバイト数 CF=1 の場合 AX=エラー・コード
	ブロック・デバイスからの制御文字列の読み取り	
	AH=44H AL=04H BL=ドライブ番号 0:カレント, 1:A… CX=制御文字列のバイト数 DS:DX=制御文字列のアドレス	CF=0 の場合 AX=読み取ったバイト数 CF=1 の場合 AX=エラー・コード
	ブロック・デバイスへの制御文字列の書き込み	
	AH=44H AL=05H BL=ドライブ番号 0:カレント, 1:A… CX=制御文字列のバイト数 DS:DX=制御文字列のアドレス	CF=0 の場合 AX=書き込んだバイト数 CF=1 の場合 AX=エラー・コード

入カステータスの検査	
AH=44H AL=06H BX=ファイル・ハンドル	CF=0 の場合 AL=00H : レディ状態でない FFH : レディ状態 CF=1 の場合 AX=エラー・コード
出カステータスの検査	
AH=44H AL=07H BX=ファイル・ハンドル	CF=0 の場合 AL=00H : レディ状態でない FFH : レディ状態 CF=1 の場合 AX=エラー・コード
ブロック・デバイスの交換可能性検査	
AH=44H AL=08H BL=ドライブ番号 0 : カレント, 1 : A...	CF=0 の場合 AX=0 : 交換可能 1 : 交換不可能 CF=1 の場合 AX=エラー・コード
リモート・ブロック・デバイスの検出	
AH=44H AL=09H BL=ドライブ番号 0 : カレント, 1 : A...	CF=0 の場合 DX=デバイス属性 CF=1 の場合 AX=エラー・コード
リモート・ハンドルの検出	
AH=44H AL=0AH BL=ドライブ番号 0 : カレント, 1 : A...	CF=0 の場合 DX=デバイス属性 CF=1 の場合 AX=エラー・コード
共用リトライ回数の設定	
AH=44H AL=0BH CX=待ち時間 DX=リトライ回数	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード



ブロック・デバイスに対する一般 IOCTL	
AH=44H AL=0DH BL = ドライブ番号 0: カレント, 1: A… CH = カテゴリ・コード CL = ファンクション・コード DS: DX = パラメータ・ブロック	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
論理ドライブ・マップの取得	
AH=44H AL=0EH BL = ドライブ番号 0: カレント, 1: A…	CF=0 の場合 AL=マッピング・コード CF=1 の場合 AX=エラー・コード
論理ドライブ・マップの設定	
AH=44H AL=0FH BL = ドライブ番号 0: カレント, 1: A…	CF=0 の場合 AL=マッピング・コード CF=1 の場合 AX=エラー・コード
IOCTL ハンドルの問い合わせ 	
AH=44H AL=10H BX = ハンドル CH = カテゴリ・コード CL = ファンクション・コード	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
IOCTL デバイスの問い合わせ 	
AH=44H AL=11H BL = ドライブ番号 0: カレント, 1: A… CH = カテゴリ・コード CL = ファンクション・コード	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード

表 3-4 ファンクション・コール(ネットワーク関連)

5EH	マシン名の取得	
	AH=5EH AL=00H DS:DX=バッファ・アドレス	CF=0 の場合 CX=ローカル・コンピュータ番号 CF=1 の場合 AX=エラー・コード
	プリンタ・セットアップ文字列の設定	
	AH=5EH AL=02H BX=割り当てインデックス番号 CX=文字列のバイト数 DS:SI=文字列のアドレス	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
5FH	プリンタ・セットアップ文字列の取得	
	AH=5EH AL=03H BX=割り当てインデックス番号 ES:DI=バッファ・アドレス	CF=0 の場合 CX=文字列のバイト数 CF=1 の場合 AX=エラー・コード
	割り当てリスト・エントリの取得	
	AH=5FH AL=02H BX=割り当てインデックス番号 DS:SI=ローカル名 ES:DI=リモート名	CF=0 の場合 BH=デバイス状況 BL=デバイス・タイプ CX=ユーザ・パラメータ CF=1 の場合 AX=エラー・コード
	割り当てリスト・エントリの作成	
	AH=5FH AL=03H BL=デバイス・タイプ CX=ユーザ・パラメータ DS:SI=ローカル名 ES:DI=リモート名	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード
	割り当てリスト・エントリの取消	
	AH=5FH AL=04H DS:SI=デバイス名のアドレス	CF=0 の場合 正常終了 CF=1 の場合 AX=エラー・コード

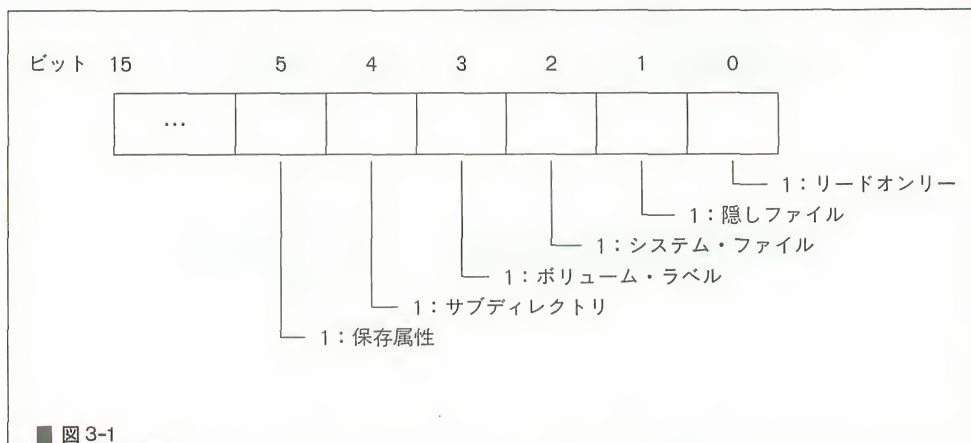


図 3-1

ファイル属性

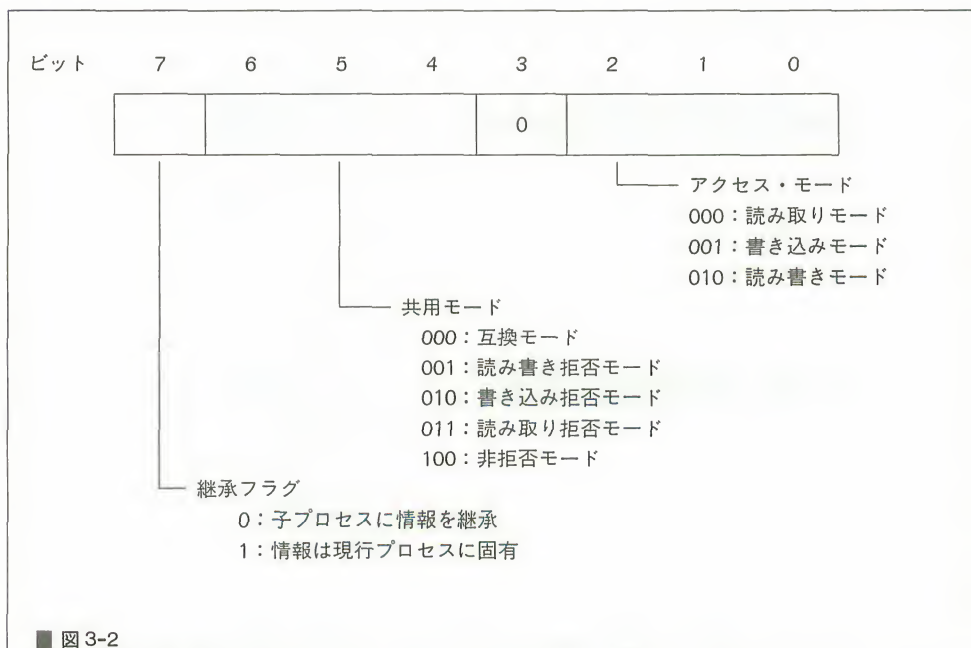


図 3-2

アクセス・モード

表 3-5 ファンクション・コールのエラー・コード

コード	意 味
00H	正常終了
01H	無効なファンクション・コード
02H	ファイルが見つからない
03H	パスが見つからない
04H	オープンされているファイルが多すぎる
05H	アクセスが拒否された
06H	無効なハンドル
07H	メモリ・コントロール・ブロックが壊れている
08H	メモリ不足
09H	無効なメモリ・ブロック・アドレス
0AH	無効な環境
0BH	無効なフォーマット
0CH	無効なアクセス・コード
0DH	無効なデータ
0EH	(未使用)
0FH	無効なドライブが指定された
10H	カレント・ディレクトリを削除しようとした
11H	同じデバイスでない
12H	これ以上ファイルがない
13H	ディスクがライト・プロテクトされている
14H	無効なディスク・ユニット
15H	ドライブの準備ができていない
16H	無効なディスク・コマンド
17H	CRC エラー
18H	リクエスト構造体の長さが無効
19H	シーク・エラー
1AH	DOS のディスクではない
1BH	セクタが見つからない
1CH	プリンタ用紙切れ
1DH	書き込み失敗
1EH	読み出し失敗
1FH	一般的な失敗
20H	シェアリング違反
21H	ロック違反
22H	無効なディスク交換
23H	FCB 使用不可
24H	シェアリング・バッファに余裕がない
25H	(未使用)
26H	ファイル操作を完了できない
27H	(未使用)
↓	
32H	ネットワークが準備できていない
33H	リモート・コンピュータが LISTEN 状態にない
34H	ネットワーク名の二重定義

35H	ネットワーク・パスが見つからない
36H	ネットワーク・ビジー
37H	ネットワーク・デバイスはこれ以上ない
38H	ネットワーク BIOS の限界を超えた
39H	ネットワーク・アダプターのハード・エラー
3AH	ネットワークからの不当な応答
3BH	予期しないネットワーク・エラー
3CH	互換性のないリモート・アダプター
3DH	プリント待ち行列がいっぱい
3EH	メモリ不足
3FH	印刷ファイルが取り消された
40H	ネットワーク名はすでに削除されている
41H	ネットワーク・アクセスが拒否された
42H	ネットワーク・デバイスのタイプが不当
43H	ネットワーク名が見つからない
44H	ネットワーク名の限界を超えた
45H	ネットワーク BIOS セッションの限界を超えた
46H	シェアリングを一時休止
47H	ネットワークの要求が受けつけられない
48H	プリンタ・ディスクのリダイレクション休止
49H	(未使用)
↓	
50H	同名のファイルがすでに存在する
51H	(未使用)
52H	ディレクトリ・エントリが作成不能
53H	割り込みタイプ 24H の失敗
54H	リダイレクションが多すぎる
55H	リダイレクションが重複している
56H	無効なパスワード
57H	無効なパラメータ
58H	ネットワーク・データ障害
59H	ネットワークによってサポートされていない機能
5AH	要求されたシステム・コンポーネントが未導入

3.3 終了アドレス(INT22H)

現在のプロセスの終了アドレスが格納されています。

DOS/V システムがプロセスを終了させる際に呼び出すので、ユーザ・プログラムから直接利用することはありません。

3.4 Ctrl-Break 割り込みアドレス(INT23H)

Ctrl + **Break** (または **Ctrl** + **C**) が押された場合に実行されます。

実際は、キーボード BIOS 側で **Ctrl** + **Break** (または **Ctrl** + **C**) を検出した直後に INT1BH が呼び出され、DOS/V の BREAK フラグが ON の場合に限り、さらにこの INT23H が呼び出されます。

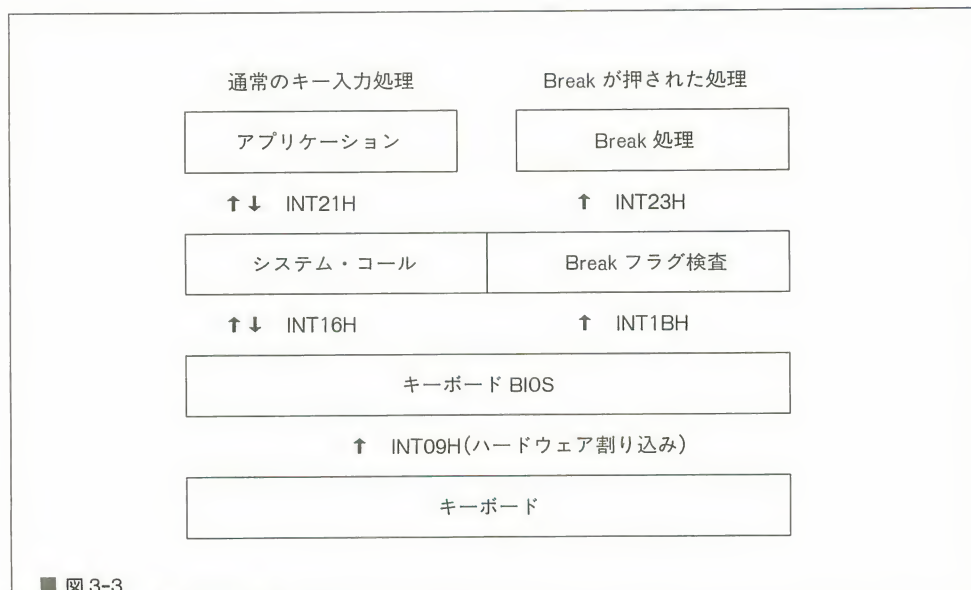


図 3-3

Ctrl-Break 割り込み処理

この割り込みは、ユーザが独自のプログラムと差し替えることが可能で、終了時のキャリーフラグの設定により、プロセスの継続・終了が制御可能です。

ただし、終了時にはすべてのレジスタが保存されていなければなりません。

INT23H 終了時のキャリーフラグの内容は、次のようになっています。

キャリーフラグ=0 : プロセスの継続

キャリーフラグ=1 : プロセスの強制終了

この割り込みをユーザ独自のプログラムに差し替えると、DOS 側の制御が効かなくなってしまうので、一般的にはファンクション・コールの「Ctrl-Break の設定」(INT21H, AH=33H)で制御を行うべきです。

3.5 重大エラー・ハンドラ(INT24H)

重大エラーとは、DOS のファンクションが実行できないときに発生するエラーで、通常はディスク・エラーに代表されるような、デバイス・ドライバの入出力エラーが主な原因です。

この割り込みは、ユーザが用意したプログラムに置き換えて制御を行うことも可能です。たとえば、ディスクがセットされていないときなどのエラー処理を、独自に行うことができます。

この割り込みが実行された直後は、次に示すレジスタに各種情報が設定されています。

AX = エラー情報

DI = エラー・コード

BP:SI = デバイス・ヘッダ制御ブロックへのポインタ

表 3-6 AX レジスタのエラー情報

ビット	エラー情報	内 容
15	エラー・デバイス	0: ディスク 1: FAT か キャラクタ・デバイス
14	未定義	
13	「無視(Ignore)」	0: 不可能 1: 可能
12	「再試行(Retry)」	0: 不可能 1: 可能
11	「中止(Abort)」	0: 不可能 1: 可能
10 ~ 9	エラーの発生領域	00: DOS/V 01: FAT 10: ディレクトリ 11: データ領域
8	エラー時の動作状態	0: 読み込み時 1: 書き込み時
7 ~ 0	エラー・ドライブ番号	0: A ドライブ 1: B ドライブ

表 3-7 DI レジスタのエラー情報

エラー・コード	意 味
00H	書き込み禁止のディスクに書き込んだ
01H	存在しないユニット番号
02H	ドライブの準備ができていない
03H	未定義コマンド
04H	データの CRC エラー
05H	リクエスト構造体の長さが違う
06H	シーク・エラー
07H	存在しないメディア・タイプ
08H	セクタが見つからない
09H	プリンタの用紙切れ
0AH	書き込みに失敗した
0BH	読み込みに失敗した
0CH	一般的なディスク不良
0DH	予約済み
0EH	予約済み
0FH	無効なディスク交換

表 3-8 BP : SI レジスタで示されるデバイス・ドライバ制御ブロック

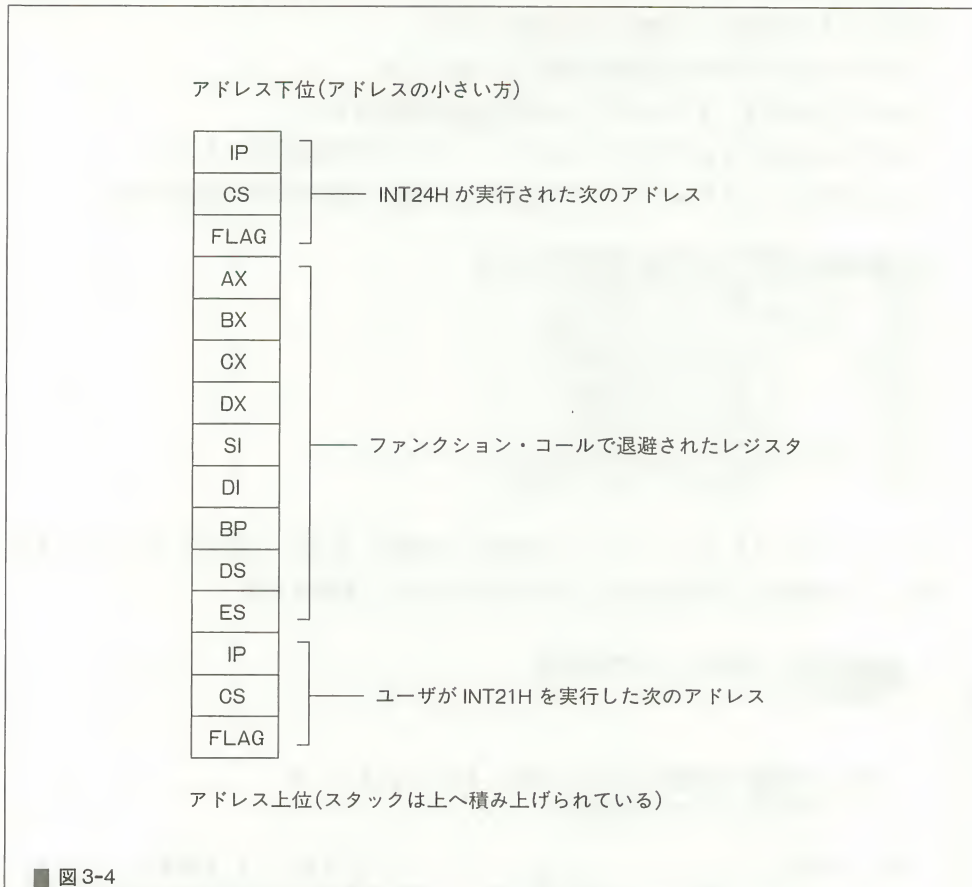
オフセット	サイズ	内 容
+0 ~+3	2 ワード	次のデバイスへのポインタ
+4 ~+5	1 ワード	デバイス属性 ビット 15=0:ブロック・デバイス 1:キャラクタ・デバイス 14=1:IOCTL ビット 3=1:CLOCK デバイス 2=1:NUL デバイス 1=1:標準出力 0=1:標準入力
+6 ~+7	1 ワード	デバイス・ストラテジ・エントリポイントへのポインタ
+8 ~+9	1 ワード	ドライバ割り込みエントリポイントへのポインタ
+A ~+11	8 バイト	キャラクタ・デバイスのファイル名 ブロック装置の場合は最初のバイトがユニット数を示す。

この割り込みからの応答は、AL レジスタに以下の値を設定して返します。

表 3-9 AL レジスタの応答内容

値	内 容
0	エラーを無視
1	操作を再試行
2	割り込み 22H を通して、プログラムを終了
3	プログラム上からのシステムコールの失敗

また、このときスタック上に保存されているレジスタ値は、次のようになっています。



したがって、もしユーザが作成したプログラムからシステム処理ルーチンに戻らずに、直接ユーザ・プログラムに戻るには、スタック上の IP, CS, FLAG を捨て AX ~ ES を POP した後に IRET すればよいことになります。

ただしこの場合、次に 0CH より高位のファンクション・コールが行われるまで DOS は不安定な状態になっているので、注意が必要です。

この割り込みを独自に置き換えるプログラムを作成する際には、以下の点に注意する必要があります。

- ・エラーがディスクにある場合は 5 回、FAT あるいはディレクトリにある場合は 3 回の試行を行うこと


```

MOV     AH, 25H
MOV     AL, 24H
MOV     DX, WORD PTR [INT24_VEC]
MOV     DS, WORD PTR [INT24_VEC+2]
INT     21H

POP     DS
RET

```

INT24:

```

PUSH    ES
PUSH    DS
PUSH    BP
PUSH    DI
PUSH    SI
PUSH    DX
PUSH    CX
PUSH    BX
PUSH    AX

```

独自のINT24ハンドラ処理

```

CMP     AL, 01H
JZ      INT24_E
MOV     AX, 0FFFFH
ADD     SP, 24

INT24_E:
MOV     BP, SP
MOV     SS:WORD PTR [BP], AX

POP     AX
POP     BX
POP     CX
POP     DX
POP     SI
POP     DI
POP     BP
POP     DS
POP     ES
IRET

```

3.6 絶対ディスクの読み書き(INT25H/INT26H)

ディスクを論理セクタ番号で、直接読み書きします。INT25H で読み込み、INT26H で書き込みを行います。

このシステム・コールでは今まで 32M バイトまでのディスクしか扱えなかったため、DOS/V ではこれを拡張しています。拡張された方式でも 32M バイト以内のディスクはアクセス可能なので、今後はこちらのシステム・コールを使用すべきでしょう。

このシステム・コールは FCB やファイル・ハンドルを介せずにアクセスを行うため、DOS/V のディスクへのアクセス方法としては最も高速ですが、これを使用すると LAN などのネットワーク・システムの仮想ドライブに対応できなくなるため、基本的には使用しないようにすべきです。

表 3-10 新しい 32M バイト以上サポートのディスクへの読み書き

入力	AL=ドライブ番号(A:0, B:1・・・) DS: BX=DISK-IO 構造体のアドレス +0～3: 読み書き開始論理セクタ番号 +4～5: 読み書きするセクタ数 +6～9: 読み書き用バッファのアドレス CX=FFFFH
出力	キャリーフラグ=0 の場合 正常終了 キャリーフラグ=1 の場合 AL=エラー・コード (INT24H の DI と同じ内容) AH=エラー状況コード 80H: 接続機構が応答障害 40H: SEEK 操作障害 20H: コントローラー・エラー 10H: データ CRC エラー 08H: DMA オーバーラン・エラー 04H: 要求されたセクタが見つからない 03H: 書き込み禁止ディスクに書き込もうとした 02H: 上記以外のエラー 01H: 無効なパラメータ

このシステム・コールにより、セグメント・レジスタを除くすべてのレジスタが破壊されます。また、復帰時には元のフラグ内容がスタックに積まれたままになっているので削除する必要がありますが、その時点でのエラー状況(キャリーフラグのこと)がフラグに設定されているので、これを破壊しないように考慮しなければなりません。

また、このシステム・コールで同時に読み出し可能なバイト数は 64K バイト以内に限り、かつ読み書き用バッファのアドレス・オフセットは FFFFH になるまでしか考慮されていないので、読み書きサイズの設定には注意が必要です。

List 3-2

絶対ディスク読み込み(拡張版)

ABS_READ:

```

PUSH    AX
PUSH    BX
PUSH    CX
PUSH    DX
PUSH    SI
PUSH    DI
PUSH    BP

MOV      AL, ドライブ番号
LDS      BX, DISKIO
MOV      CX, 0FFFFH      ; 拡張モード指定
INT      25H
POP      CX                ; スタック上のフラグの削除
JB       ERROR            ; エラー処理へ

POP      BP
POP      DI
POP      SI
POP      DX
POP      CX
POP      BX
POP      AX
RET

```

ERROR: エラー処理ルーチン

DISKIO

```

DD      読み書き開始論理セクタ番号
DW      読み書きセクタ数
DD      読み書きバッファアドレス

```

表 3-11 従来の 32M バイト以内のディスクへの読み書き(参考)

入力	AL=ドライブ番号(A:0, B:1・・・) DS:BX=読み書き用バッファのアドレス CX=読み書きするセクタ数 DX=読み書き開始論理セクタ番号
出力	キャリーフラグ=0 の場合 正常終了 キャリーフラグ=1 の場合 AL=エラー・コード (INT24H の DI と同じ内容) AH=エラー状況コード

3.7 プログラムの常駐終了 (INT27H)

このシステム・コールは使用すべきではありません。

このシステム・コールでは 64K バイト以内のプロセスの終了を行いますが、CP/M との互換性を保つために残っているだけなので、DOS の Ver.2.00 以降は以下のファンクションを使用すべきです。

常駐のままプロセス終了 : INT21H, AH=31H

3.8 バックグラウンド処理(INT28H)[非公開]

バックグラウンド・タスクの実行を行います。実際にはファンクション・コール(INT21H)の機能番号 01H から 0CH を実行中に、キー入力待ちなどの時間待ちとして、このシステムコールが実行されます。たとえば DOS/V コマンドの PRINT.EXE などはこの機能を利用してスプーラ機能を実現しています。

したがって、このシステム・コールへアプリケーション独自の処理を設定することでバックグラウンド処理が実現できますが、次の点に注意しなければなりません。

- ・他のバックグラウンド処理を実行するために、元の INT28H の処理を忘れずにチェーン(far コール)しておかなければならない
- ・この割り込みはファンクション・コールの 01H から 0CH が呼び出されたときに実行されるので、これらのファンクションが必ず定期的に実行されるように考慮しておかなければならない
- ・割り込み中は、ファンクション・コール 01H から 0CH は実行できない
- ・割り込み中は DOS/V の内部スタックを使用しているので、スタックは独自のものを用意すべきである
- ・この割り込みでは、CS : IP 以外のすべてのレジスタは破壊される

List 3-3

割り込みの設定例

```
CHAIN:                                     ; 割り込みのチェーン
      MOV     AH, 35H
      MOV     AL, 28H
      INT     21H
      MOV     WORD PTR [OLD_PROC], BX
      MOV     WORD PTR [OLD_PROC+2], ES
                                     ; 割り込みの設定
      MOV     AH, 25H
      MOV     AL, 28H
      LEA     DX, NEW_PROC

      RET
```

NEW_PROC: ; 追加された処理

```
MOV CS:WORD PTR [OLD_STACK], SP
MOV CS:WORD PTR [OLD_STACK+2], SS
MOV SP, CS
MOV SS, SP
LEA SP, NEW_STACK+512
```

```
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH SI
PUSH DI
PUSH BP
PUSH DS
PUSH ES
```

独自の割り込み処理

```
POP ES
POP DS
POP BP
POP DI
POP SI
POP DX
POP CX
POP BX
POP AX
```

```
MOV SP, CS:WORD PTR [OLD_STACK]
MOV SS, CS:WORD PTR [OLD_STACK+2]
JMP FAR PTR CS:[OLD_PROC]
```

OLD_PROC	DD	?	; 元のINT28H
OLD_STACK	DD	?	; 元のスタック
NEW_STACK	DB	512 DUP (?)	; 独自スタック

3.9 高速 1 文字出力(INT29H)[非公開]

DOS/V の中で最も高速な文字出力が、このシステム・コールです。標準のコンソール出力はすべてこのシステム・コールを利用しています。ANSI.SYS が組み込まれている場合は、エスケープ・シーケンスまでサポートされますが、DOS/V のリダイレクト機能はこれ以前の段階で実現されているため使えなくなります。

入力	AL=出力する文字
出力	なし

3.10 コマンドの起動(INT2EH)[非公開]

このシステム・コールは使用すべきではありません。

このシステム・コールでは、指定された文字列を COMMAND.COM へ渡して処理させます。しかし、全レジスタが破壊される、多重コールがかかった場合にプロセスが常駐してしまう、など問題が多いため、基本的にはファンクション・コールの「プログラムの実行」(INT21H, AH=4BH)を使用すべきです。

ただし、このシステム・コールにより環境変数を操作した場合は、マスタの環境(COMMAND.COM のもっている環境変数)が操作可能なので利用価値はあるかもしれません。

入力	DS:SI=コマンド文字列
出力	AL=リターン・コード
CS 以外のすべてのレジスタが破壊される。 なお、コマンド文字列は以下の構成となっている。	
1 バイト目	終端の CR(0DH)を除いた文字数
2 バイト目以降	コマンド・ライン文字列
終端	CR(0DH)

3.11 多重割り込み(INT2FH)

このシステム・コールは、常駐プロセスとのインタフェイスを行うもので、DOS/V というよりも DOS5 から新設されたものです。これにより各種常駐プロセスとのやり取りがより容易になっています。

基本的には AH レジスタで多重番号(プロセスの番号)を、AL レジスタでそれぞれのプロセスの機能を指定します。

多重番号は 00H から BFH まだが DOS/V で予約されており、C0H から FFH まだがアプリケーションに開放されていますが、残念ながらこの多重番号の予約方法までは規定されていないので、アプリケーション側でも変更が効くように設計すべきです。

また、すべての多重番号の機能番号 00H は「インストール・ステータスの取得」を返すことが規定されており、これによりその常駐プロセスの状況を知ることができます。アプリケーション側では、その機能の利用に先だって、このステータス进行检查しておく必要があります。

表 3-12 常駐プロセスのインストール・ステータス

AL の値	意 味
00H	常駐していない
01H	常駐できないプロセスが存在している
FFH	すでに常駐している

表 3-13 DOS/V で使用している多重番号

多重番号	常駐プロセス名
01H	PRINT.EXE
06H	ASSIGN.COM
10H	SHARE.EXE
11H	ネットワーク
12H	PCDOS.SYS
14H	NLSFUNC.EXE
16H	アイドル・コールおよび DPMI
1AH	ANSI.SYS
43H	HIMEM.SYS
48H	DOSKEY.SYS
4AH	ハイ・メモリ
4BH	タスク・スイッチャー
ADH	KEYB.COM
B0H	GRAFTABL.COM
B7H	APPEND.EXE

以下、DOS/V システム側で常駐しているプロセスのうち、インストール・ステータス以外の機能をもつものを取り上げておきます。

(1) PRINT.EXE(多重番号 01H)

機能番号 AL=	内 容
01H	ファイルの待ち行列への追加 <input type="text" value="入力"/> DS:DX=ファイル名文字列(先頭と終端が 00H) <input type="text" value="出力"/> AX=エラー・コード
02H	待ち行列からのファイルの削除 <input type="text" value="入力"/> DS:DX=ファイル名文字列(先頭と終端が 00H) <input type="text" value="出力"/> AX=エラー・コード
03H	すべての待ち行列の取消 <input type="text" value="出力"/> AX=エラー・コード
04H	印刷の一時保留と状況の取得 <input type="text" value="出力"/> AX=エラー・コード DX=エラー回数(障害がなければ 0) DS:DI=待ち行列リスト(各エントリは 64 バイト、終端は先頭が 00H)
05H	印刷の再開 <input type="text" value="出力"/> AX=エラー・コード
06H	プリンタ・デバイス・ヘッダのアドレスの取得 <input type="text" value="出力"/> DS:SI=プリンタ・デバイス・ヘッダのアドレス

なお、返されるエラー・コードは PRINT.EXE 独自のものです。次の意味をもちます。

エラー・コード	意 味
0001H	無効な機能を指定した
0002H	指定されたファイルが見つからない
0003H	無効なパス名が指定された
0004H	オープンしたファイル数が多すぎる
0005H	アクセスが拒否された
0006H	待ち行列がいっぱい
0009H	スプーラが BUSY 状態
000CH	ファイル名が 64 バイトを超えている
000FH	無効なドライブが指定されている

(2) アイドル・コール(多重番号 16H)

アプリケーションがアイドル状態(ループ待ちしている状態)であることを DOS/V に通知します。

これはマルチタスクでのタスク制御のための機能で、現在の DOS/V はマルチタスクをサポートしていないため、この機能をコールしてもすぐに戻ってくるだけです。したがって、現状ではまったく無意味な機能ですが、DOS/V や Windows が将来マルチタスクをサポートする際に必要になります。ですから、アプリケーションがアイドル状態にあるときは、積極的にこの機能をコールすべきです。

機能番号 AL=	内 容
80H	アイドル状態の通知 <div>出力</div> AL=80H: マルチタスクはサポートされていない 00H: マルチタスクがサポートされている

(3) HIMEM.SYS(多重番号 43H)

機能番号 AL=	内 容
10H	HIMEM.SYS のエントリ・アドレスの取得 <div>出力</div> ES: BX=エントリ・アドレス

(4) DOSKEY.COM(多重番号 48H)

機能番号 AL=	内 容
10H	コマンド・ラインの取得 <div>入力</div> DS: DX=コマンド・ライン・バッファへのアドレス <div>出力</div> AX=終了ステータス(0000H で正常終了)

マクロ名が入力されたときはいったん正常終了しますが、再度この機能を呼び出すことで展開された内容が取得できます。

また、コマンド・ライン・バッファは 128 バイトで以下の構成となっています。

オフセット	内 容
00H	128(バッファのバイト数で固定値)
01H	CR(0DH)を除く、取得されたバイト数
02H	入力バッファ
↓	
7FH	

(5) ハイ・メモリ (多重番号 4AH)

機能番号 AL =	内 容
01H	<p>HMA 領域の空き容量の取得</p> <p>出力 BX = HMA 空きバイト数</p> <p>ES : DI = HMA 空き領域先頭アドレス</p> <p>DOS/V が HMA を使用していないときは、次の値が返る</p> <p>BX = 0000H</p> <p>ES : DI = FFFF : FFFFH</p>
02H	<p>HMA 領域の割り当て</p> <p>入力 BX = 割り当てバイト数</p> <p>出力 ES : DI = 割り当てられた HMA 先頭アドレス</p> <p>(FFFF : FFFFH ならば失敗)</p>

(6) タスク・スイッチャー (多重番号 4BH)

機能番号 AL =	内 容
01H	<p>通知ファンクション・ハンドラのリンク・リストの作成</p> <p>入力 ES : BX = 0000 : 0000H</p> <p>CX : DX = サービス・ファンクション・ハンドラ</p> <p>出力 ES : BX = コール・バック情報構造体のアドレス</p> <p>(0 ならばクライアントに通知の必要なし)</p>
02H	<p>サービス・ファンクション・ハンドラの取得</p> <p>出力 ES : DI = サービス・ファンクション・ハンドラ</p>
03H	<p>タスク・スイッチャ ID の取得</p> <p>入力 BX = 0000H</p> <p>ES : DI = サービス・ファンクション・ハンドラ</p> <p>出力 BX = スイッチャ ID (0000H ならば割り当て不可)</p>
04H	<p>タスク・スイッチャ ID の開放</p> <p>入力 BX = スイッチャ ID</p> <p>ES : DI = サービス・ファンクション・ハンドラ</p> <p>出力 BX = ステータス (0000H ならば正常終了)</p>
05H	<p>タスク・スイッチャのインスタンス・データの識別</p> <p>入力 ES : BX = 0000 : 0000H</p> <p>CX : DX = サービス・ファンクション・ハンドラ</p> <p>出力 ES : BX = スタートアップ構造体のアドレス</p> <p>(0000H ならばデータをもっていない)</p>

(7) KEYB.COM(多重番号 ADH)

機能番号 AL =	内 容
80H	バージョン番号の取得 <div>出力</div> BH = メジャー・バージョン番号 BL = マイナー・バージョン番号
81H	アクティブ・コード・ページの設定 <div>入力</div> BX = コード・ページ <div>出力</div> キャリーフラグが 0 ならば正常終了
82H	国別フラグの設定 <div>入力</div> BX = 国別フラグ 00H : US キーボード FFH : US キーボード以外 <div>出力</div> キャリーフラグ = 0 ならば正常終了
83H	国別フラグの取得 <div>出力</div> BX = 国別フラグ 00H : US キーボード FFH : US キーボード以外

(8) APPEND.EXE(多重番号 B7H)

機能番号 AL =	内 容
02H	バージョン・フラグの取得 <div>出力</div> AX = ステータス (FFFFH ならば DOS5 と互換)
04H	活動中の APPEND パスの取得 <div>出力</div> ES : DI = ディレクトリ・リスト・アドレス
06H	動作モードの取得 <div>出力</div> BX = 動作モード 0001H : APPEND が使用可能 8000H : /X がオン 4000H : /PATH がオン 2000H : /E がオン
07H	動作モードの設定 <div>入力</div> BX = 動作モード
11H	フル・パス名設定フラグの設定

第 4 章

BIOS コール

「BIOS」は、DOS/V のシステムコールよりもさらに下位に位置し、よりハードウェアに密着したシステム・サービスです。

これよりも下位レベルの操作を行いたければ、あとはハードウェアを直接操作するしかありませんが、一般のアプリケーションが利用するのはこの BIOS レベルまでにとどめておくべきです。

実際のところ、いかに互換性の高いといわれる IBM-PC であっても、ハードウェアレベルでは機種ごとに若干の違いがあります。これを吸収するために BIOS が用意されているのですから、テキスト表示をグラフィックスでシミュレートしている DOS/V にあっても、これより下位の操作を行って、互換性の高い日本語表示を行うのは困難といわざるをえないでしょう。

IBM-PC の BIOS を表 4-1 に示します。

表 4-1 IBM-PC の BIOS 一覧

OADG 共通対応	割り込み	内 容
○	INT10H	ディスプレイ
	INT11H	装置構成情報
	INT12H	メモリ・サイズ取得
	INT13H	ディスク
	INT14H	RS-232C
	INT15H	システム・サービス
○	INT16H	キーボード
○	INT17H	プリンタ
	INT1AH	タイマ・クロック
○	INT33H	マウス

このうち、OADG で共通に規定されているのは表 4-1 中に○のあるものだけです。それ以外の BIOS は、「AT 互換機、PS/2、PS/55 などの間で完全な互換性がないので、通常のアプリケーションでは利用すべきでない」とされています。

ですから、アプリケーションは互換性確保のために、できる限りこの範囲で機能を実現すべきですが、現実的にはこれら以外の BIOS を利用しなければ実現できない機能も多くあります。本章ではそれらの機能を、その対処法とともに取り上げます。

また、表 4-2 に示すものは BIOS ではありませんが、BIOS と関連が深く、利用価値の高いものなので同時に取り上げます。

表 4-2 BIOS 関連の割り込み

割り込み	内 容
INT05H	ハードコピー
INT1BH	Ctrl-Break 割り込み
INT1CH	インターバル・タイマ割り込み

4.1 BIOS のワークエリア

IBM-PC では BIOS のワークエリアが公開されており、ここから各種の重要な情報を取得することができます。IBM-PC の世界では、このワークエリアを参照することは「常識的なこと」なのですが、完全な互換性が確保できていないために、DOS/V ではあまり推奨されていません(というより公開されていません)。

しかし、ワークエリアでなければ取得できない情報も多いので、参照せざるをえないのが現状です。

BIOS ワークエリアはセグメント 0040H より、DOS/V のワークエリアはセグメント 0050H より格納されています。

表 4-3 BIOS のワークエリア一覧

セグメント 0040H

オフセット	サイズ	内 容	公開
0000H	4 ワード	RS-232C ポート・アドレス 0000H : COM1 0002H : COM2 0004H : COM3 0006H : COM4	○
0008H	4 ワード	プリンタ・ポート・アドレス 0008H : LPT1 000AH : LPT2 000CH : LPT3 000EH : 予約済み	○
0010H	1 ワード	システム構成情報 ビット 15 ~ 14 : プリンタ・ポート数 ビット 13 : 内蔵モデムの有無 ビット 11 ~ 9 : RS-232SC ポート数 ビット 7 ~ 6 : 接続 FDD 数-1 ビット 5 ~ 4 : ビデオ・タイプ 01 : 40×25 カラー 10 : 80×25 カラー 11 : 80×25 モノクロ ビット 2 : マウスの有無 ビット 1 : 演算コプロセッサの有無 ビット 0 : IPL ディスク(つねに 1)	
0012H	1 バイト	予約済み	
0013H	1 ワード	メモリ・サイズ	
0015H	1 バイト	予約済み	

0017H	1 バイト	キーボード・シフト・ステータス (PC84 キーボード用) ビット 7: Insert ビット 6: CapsLock ビット 5: NumLock ビット 4: ScrollLock ビット 3: Alt ビット 2: Ctrl ビット 1: 左 Shift ビット 0: 右 Shift	
0018H	1 バイト	キーボード・シフト・ステータス (AT101 拡張キーボード用) ビット 7: Insert ビット 6: CapsLock ビット 5: NumLock ビット 4: ScrollLock ビット 3: Pause ビット 2: SysRq ビット 1: 左 Alt ビット 0: 左 Ctrl	
0019H	1 バイト	予約済み	
001AH	1 ワード	キーボード・バッファ読み出し位置ポインタ	
001CH	1 ワード	キーボード・バッファ書き込み位置ポインタ	
001EH	16 ワード	キーボード・バッファ	
003EH	1 バイト	FDD リキャリプレート・フラグ ビット 3: ドライブ 3 ビット 2: ドライブ 2 ビット 1: ドライブ 1 ビット 0: ドライブ 0	
003FH	1 バイト	FDD モータ ON フラグ ビット 3: ドライブ 3 ビット 2: ドライブ 2 ビット 1: ドライブ 1 ビット 0: ドライブ 0	
0040H	1 バイト	FDD モータ OFF タイマ・カウンタ (55ms 単位)	
0041H	1 バイト	FDD エラー・ステータス	○
0042H	7 バイト	FDD コントローラ・ステータス	
0049H	1 バイト	ビデオ・モード番号	
004AH	1 ワード	画面 1 行当たりの文字数	
004CH	1 ワード	1 ページのバイト数	
004EH	1 ワード	ビデオ RAM 開始オフセット・アドレス	
0050H	16 バイト	画面ページごとのカーソル位置 0050H: ページ 0 桁位置 0051H: ページ 0 行位置 0052H ~ 未使用	

0060H	2 バイト	カーソルの高さ 0060H : エンド位置 0061H : スタート位置	
0062H	1 バイト	現在表示されているページ番号(つねに 0)	
0063H	1 ワード	CRT コントローラのポート・アドレス カラー : 03D4H モノクロ : 03B4H	
0065H	7 バイト	予約済み	
006CH	2 ワード	マスタ・クロック・カウンタ	
0070H	1 バイト	クロック・オーバーフロー・カウンタ	
0071H	1 バイト	Ctrl-Break フラグ ビット 7=0 : OFF =1 : ON	
0072H	1 ワード	リセット・フラグ 電源投入時のメモリ・チェックの後に 1234H に設定される。	
0074H	1 バイト	ハードディスクのエラー・ステータス	
0075H	1 バイト	ハードディスクのドライブ数	
0076H	1 バイト	ハードディスク制御バイト	
0077H	1 バイト	ハードディスク・ポート・オフセット	
0078H	4 バイト	プリンタのタイムアウト時間 0078H : LPT1 0079H : LPT2 007AH : LPT3 007BH : LPT4	
007CH	4 バイト	RS-232C のタイムアウト時間 007CH : COM1 007DH : COM2 007EH : COM3 007FH : COM4	
0080H	1 ワード	キーボード・バッファ先頭オフセット(通常 001EH)	
0082H	1 ワード	キーボード・バッファ終端オフセット(通常 003EH)	
0084H	1 バイト	1 画面の行数-1	
0085H	1 ワード	1 文字の縦のビット数	
0087H	1 ワード	EGA 制御情報	
0089H	1 バイト	ビデオ・ディスプレイ・データ	
008AH	1 バイト	ビデオ・インデックス	
008BH	11 バイト	ディスクの制御ステータス情報	

0096H	1 バイト	キーボード・モード・フラグ ビット 7: ReadID コマンド実行中 ビット 6: キーボード ID 第 1 バイト取得中 ビット 4: 101/102 キーボード ビット 3: 右 Alt ビット 2: 右 Ctrl ビット 1: 最後のコードが E0 ビット 0: 最後のコードが E1	
0097H	1 バイト	キーボード LED ステータス ビット 7: キーボード転送エラー ビット 6: モード・インジケータ更新 ビット 5: Resend コード受信 ビット 4: ACK コード受信 ビット 2: 右 Ctrl ビット 1: CapsLock LED ビット 0: ScrollLock LED	
0098H	9 バイト	リアルタイム・クロック制御情報	
00A8H	2 ワード	ビデオ・パラメータ・テーブルのアドレス	

表 4-4 DOS/V のワークエリア一覧

セグメント 0050H

オフセット	サイズ	内 容	公開
0000H	1 バイト	プリント・スクリーン・フラグ 00H = ハードコピー未使用または終了 01H = ハードコピー実行中 FFH = エラー発生	○
0004H	1 バイト	1 ドライブでの 2 ドライブ・シミュレート状況 00H = ドライブ A として動作中 01H = ドライブ B として動作中	○
0022H	14 バイト	FORMAT コマンド作業領域	
0030H	4 バイト	MODE コマンド作業領域	
0034H ↓ 00FFH		予約済み	

4.2 ビデオ BIOS(INT10H)

ビデオ BIOS はディスプレイ BIOS とも呼ばれ、IBM-PC の数ある BIOS の中でも最も特徴的なものです。

IBM-PC ではもともと、ビデオ・カードがオプションであったこともあり、実際にはこのビデオ BIOS は、登載されたビデオ・カード上に存在しています(一部、本体組み込み済みのものもあります)。したがって、ビデオ BIOS 自体はビデオ・カードの交換とともに入れ替わるという、実に巧妙かつ合理的な方式が実現されています。DOS/V では、このビデオ BIOS をさらにソフトウェア的に拡張することで、日本語表示を実現しています。

また、最近では V-Text と呼ばれる高解像度対応のビデオ・モードも日本 IBM によって正式に規格化され、ビデオ BIOS はさらに複雑化してきています。

表 4-5 ビデオ BIOS(INT10H)機能一覧

機能番号	機能内容
00H	ビデオ・モードの設定
01H	カーソル形状の設定
02H	カーソル位置の設定
03H	カーソル位置の取得
05H	アクティブ・ページの選択
06H	上方向へのスクロール
07H	下方向へのスクロール
08H	カーソル位置の文字と属性の読み取り
09H	カーソル位置の文字と属性の書き込み
0AH	カーソル位置の文字の書き込み
0CH	ドットの書き込み
0DH	ドットの読み取り
0EH	テレタイプ式書き込み
0FH	ビデオ・モードの取得
1000H	パレット・レジスタの設定
1001H	オーバースキャン・レジスタの設定
1002H	パレット・レジスタの一括設定
1007H	パレット・レジスタの読み取り
1008H	オーバースキャン・レジスタの読み取り
1009H	パレット・レジスタの一括読み取り
1010H	カラー・レジスタの設定
1012H	カラー・レジスタの一括設定

1015H	カラー・レジスタの読み取り
1017H	カラー・レジスタの一括読み取り
1100H	ユーザ定義の文字フォント登録
1118H	高密度文字フォントへの切り替え (V-Text)
1130H	画面の行数情報 (V-Text)
1131H	拡張モード・テーブルの読み取り (V-Text)
12-10H	VGA 情報の取得(非公開)
12-20H	プリント・スクリーン処理の切り替え
12-31H	モード変更時のパレット・ロード設定
12-32H	ビデオ・メモリへのアクセス設定
12-33H	グレースケールの設定
12-34H	カーソル・エミュレーションの設定
12-36H	ビデオ・スクリーンの ON/OFF
12-38H	文字フォント・サイズの変更 (V-Text)
12-39H	表示文字密度の変更 (V-Text)
12-3AH	拡張ビデオ・モード設定情報の取得 (V-Text)
1300H	文字列の書き込み(1)
1301H	文字列の書き込み(2)
1302H	文字列の書き込み(3)
1303H	文字列の書き込み(4)
1310H	文字ブロックの読み取り(1)
1311H	文字ブロックの読み取り(2)
1320H	文字ブロックの書き込み(1)
1321H	文字ブロックの書き込み(2)
1800H	フォント・パターンを読み書き(1)
1801H	フォント・パターンを読み書き(2)
1A00H	ディスプレイ組み合わせコードの読み取り
1DH	キーボード・シフト標識域の制御
FEH	ビデオ・バッファ・アドレスの読み取り
FFH	画面表示の更新

以下、ビデオ BIOS(INT10H)の機能を各番号別に示します。

▶ INT10H(00H) ビデオ・モードの設定	
入力パラメータ AH=00H AL=ビデオ・モード	リターン情報 なし
機 能 ビデオ・モードを設定します。 AL の第 7 ビットが 1 の場合は画面消去を行いません。	

表 4-6 DOS/V の日本語ビデオ・モード

モード	タイプ	解像度	文字数	色数
03H	文字モード	640×475 *	80×25	16
11H	グラフィック・モード	640×480	80×30	2
12H	グラフィック・モード	640×480	80×30	16
70H	V-Text 文字モード	可変	可変	16
71H	V-Text 拡張文字モード	可変	可変	16
72H	グラフィック・モード	640×480	80×25	16
73H	拡張文字モード	640×475 *	80×25	16

※ V-Text モードは、V-Text 対応ドライバがインストールされていない
 ければ利用できない

*がついている解像度は、V-Text 時は可変





表 4-7 各ビデオ・モード時の文字属性

モード	属 性	
03H 70H	<div> <div>70</div> <div>BI BR BG BB CI CR CG CB</div> </div>	BI : 背景色の輝度 BR : 背景色の赤 BG : 背景色の緑 BB : 背景色の青 CI : 文字色の輝度 CR : 文字色の赤 CG : 文字色の緑 CB : 文字色の青
71H 73H	<div> <div>70</div> <div>BI BR BG BB CI CR CG CB</div> </div> 属性バイト 0 <div> <div>1508</div> <div>UL 0 0 0 VK HK 0 0</div> </div> 属性バイト 1 <div> <div>2316</div> <div>0 0 0 0 0 0 0 0</div> </div> 属性バイト 2	BI : 背景色の輝度 BR : 背景色の赤 BG : 背景色の緑 BB : 背景色の青 CI : 文字色の輝度 CR : 文字色の赤 CG : 文字色の緑 CB : 文字色の青 UL : アンダーライン VK : 縦野線 HK : 横野線
11H	<div> <div>70</div> <div>XR 0 0 0 0 0 0 0</div> </div>	XR : 現行内容と XOR

12H 72H	7	0						XR: 現行内容と XOR CI: 文字色の輝度 CR: 文字色の赤 CG: 文字色の緑 CB: 文字色の青
	XR	0	0	0	CI	CR	CG	CB

▶ INT10H(01H) カーソル形状の設定

入力パラメータ	リターン情報
AH=01H CH=カーソルの開始行 CL=カーソルの終了行	なし

機 能					
カーソルの形状を設定します。					
カーソルの形状については、以下の形状が推奨されています。					
名称	底形	上部	下部	箱形	オフ
CH	06H	00H	04H	00H	20H
CL	07H	03H	07H	07H	00H
形状					

▶ INT10H(02H) カーソル位置の設定

入力パラメータ AH=02H BH=00H(ページ番号) DH=行位置 DL=桁位置	リターン情報 なし
機能 カーソルの位置を設定します。 座標は半角単位で、左上を(0, 0)とします。	

▶ INT10H(03H) カーソル位置の取得	
入力パラメータ AH=03H BH=00H(ページ番号)	リターン情報 DH=行位置 DL=桁位置 CH=カーソルの開始行 CL=カーソルの終了行
機能 カーソルの位置を取得します。	

▶ INT10H(05H) アクティブ・ページの選択	
入力パラメータ AH=05H AL=00H(ページ番号)	リターン情報 なし
機能 アクティブ・ページを指定します。日本語モードにおけるページは0のみなので、この機能は実際には使用しません。	

▶ INT10H(06H) 上方向へのスクロール	
入力パラメータ AH=06H AL=移動行数 BH=スクロール後の空白属性 CH=左上の行位置 CL= 桁位置 DH=右下の行位置 DL= 桁位置	リターン情報 なし
機能 指定範囲を上方向へスクロールし、空いた行を BH の属性で消去します。 AL=00H の場合は指定範囲を BH の属性で消去するので、画面消去への応用も可能です。	

▶ INT10H(07H) 下方向へのスクロール	
入力パラメータ AH=07H AL = 移動行数 BH=スクロール後の空白属性 CH=左上の行位置 CL = 桁位置 DH=右下の行位置 DL = 桁位置	リターン情報 なし
機 能 指定範囲を下方向へスクロールし、空いた行を BH の属性で消去します。 AL=00H の場合は指定範囲を BH の属性で消去するので、画面消去への応用も可能です。	

▶ INT10H(08H) カーソル位置の文字と属性の読み取り	
入力パラメータ AH=08H BH=00H(ページ番号)	リターン情報 AH=属性 AL=文字コード
機 能 カーソル位置の文字コードと属性を読み取ります。	

▶ INT10H(09H) カーソル位置の文字と属性の書き込み	
入力パラメータ AH=09H AL = 文字コード BH=00H(ページ番号) BL = 属性 CX = 書き込む文字数	リターン情報 なし
機 能 カーソル位置へ文字コードと属性を CX 回だけ書き込みます。 この書き込みによっても、カーソル位置は移動しません。全角文字を書き込む場合は、第1バイト目の書き込みの直後にカーソル位置を移動させ、第2バイト目を書き込む必要があります。	

▶ INT10H(OAH) カーソル位置の文字の書き込み	
入力パラメータ AH=0AH AL=文字コード BH=00H(ページ番号) CX=書き込む文字数	リターン情報 なし
機 能 カーソル位置へ、文字コードを CX 回だけ書き込みます。 この書き込みによっても、カーソル位置は移動しません。全角文字を書き込む場合は、第1バイト目の書き込みの直後にカーソル位置を移動させ、第2バイト目を書き込む必要があります。 グラフィック・モード時には、この機能は使用しないでください。	

▶ INT10H(OCH) ドットの書き込み

入力パラメータ AH=0CH AL=カラー・コード BX=00H(ページ番号) CX=X 座標 DX=Y 座標	リターン情報 なし
---	-------------------------

機 能

指定座標へ AL で示されるカラー・コードで、ドットを書き込みます。
この機能はグラフィック・モードでのみ有効です。

・2色モード時のカラー・コード

7								0
X	0	0	0	0	0	0	0	I

X: 現行内容と XOR

I: 輝度

・16色モード時のカラー・コード

7							0
X	0	0	0	I	R	G	B

X: 現行内容と XOR

I: 輝度

R: 赤

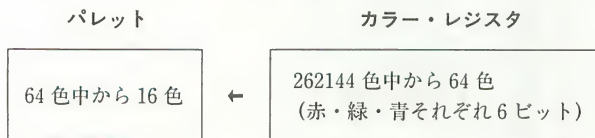
G: 緑

B: 青

▶ INT10H(0DH) ドットの読み取り																		
入力パラメータ AH=0DH CX=X 座標 DX=Y 座標	リターン情報 AL=カラー・コード なし																	
機 能 指定座標のドットのカラー・コードを読み込みます。 この機能はグラフィック・モードでのみ有効です。 ・2色モード時のカラー・コード <table border="1" style="margin-left: 40px;"><tr><td style="text-align: center;">7</td><td colspan="6"></td><td style="text-align: center;">0</td></tr><tr><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td></tr></table> <div style="margin-left: 400px;">I : 輝度</div>		7							0	0	0	0	0	0	0	0	0	1
7							0											
0	0	0	0	0	0	0	0	1										
・16色モード時のカラー・コード <table border="1" style="margin-left: 40px;"><tr><td style="text-align: center;">7</td><td colspan="4"></td><td colspan="3"></td><td style="text-align: center;">0</td></tr><tr><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">I</td><td style="text-align: center;">R</td><td style="text-align: center;">G</td><td style="text-align: center;">B</td></tr></table> <div style="margin-left: 400px;">I : 輝度 R : 赤 G : 緑 B : 青</div>		7								0	0	0	0	0	I	R	G	B
7								0										
0	0	0	0	I	R	G	B											

※ カラーコードと表示色の関係

DOS/V で採用した VGA では、二重のカラー構造をもっています。



これらの初期値は以下のとおりです。

表 4-8 パレットの初期値

パレット 番号	カラー・ レジスタ	表示色	パレット 番号	カラー・ レジスタ	表示色
00H	00H	黒	08H	38H	灰色
01H	01H	青	09H	39H	薄い青
02H	02H	緑	0AH	3AH	薄い緑
03H	03H	水色	0BH	3BH	薄い水色
04H	04H	赤	0CH	3CH	薄い赤
05H	05H	紫	0DH	3DH	薄い紫
06H	14H	茶色	0EH	3EH	薄い黄色
07H	07H	白	0FH	3FH	明るい白

表 4-9 カラー・レジスタの初期値

レジスタ番号	赤階調	緑階調	青階調	表示色
00H	00H	00H	00H	黒
01H	00H	00H	2AH	青
02H	00H	2AH	00H	緑
03H	00H	2AH	2AH	水色
04H	2AH	00H	00H	赤
05H	2AH	00H	2AH	紫
06H	2AH	2AH	00H	黄色
07H	2AH	2AH	2AH	白
08H	00H	00H	15H	暗い青
09H	00H	00H	3FH	明るい青
0AH	00H	2AH	15H	
0BH	00H	2AH	3FH	緑青
0CH	2AH	00H	15H	
0DH	2AH	00H	3FH	
0EH	2AH	2AH	15H	
0FH	2AH	2AH	3FH	
10H	00H	15H	00H	暗い緑
11H	00H	15H	2AH	
12H	00H	3FH	00H	明るい緑
13H	00H	3FH	2AH	空色
14H	2AH	15H	00H	茶色
15H	2AH	15H	2AH	
16H	2AH	3FH	2AH	
17H	2AH	00H	00H	
18H	00H	15H	15H	暗い水色
19H	00H	15H	3FH	
1AH	00H	3FH	15H	青緑
1BH	00H	3FH	3FH	明るい水色
1CH	2AH	15H	15H	
1DH	2AH	15H	3FH	
1EH	2AH	3FH	15H	
1FH	2AH	3FH	3FH	
20H	15H	00H	00H	暗い赤
21H	15H	00H	2AH	
22H	15H	2AH	00H	黄緑
23H	15H	2AH	2AH	
24H	3FH	00H	00H	明るい赤
25H	3FH	00H	2AH	濃いピンク
26H	3FH	2AH	00H	やまぶき色
27H	3FH	2AH	2AH	
28H	15H	00H	15H	暗い紫
29H	15H	00H	3FH	すみれ色
2AH	15H	2AH	15H	
2BH	15H	2AH	3FH	

2CH	3FH	00H	15H	紫赤
2DH	3FH	00H	3FH	明るい紫
2EH	3FH	2AH	15H	
2FH	3FH	2AH	3FH	
30H	15H	15H	00H	暗い黄色
31H	15H	15H	2AH	
32H	15H	3FH	00H	
33H	15H	3FH	2AH	
34H	3FH	15H	00H	
35H	3FH	15H	2AH	
36H	3FH	3FH	00H	明るい黄色
37H	3FH	3FH	2AH	
38H	15H	15H	15H	灰色
39H	15H	15H	3FH	薄い青
3AH	15H	3FH	15H	薄い緑
3BH	15H	3FH	3FH	薄い水色
3CH	3FH	15H	15H	薄い赤
3DH	3FH	15H	3FH	薄い紫
3EH	3FH	3FH	15H	薄い黄色
3FH	3FH	3FH	3FH	明るい白

▶ INT10H(0EH) テレタイプ式書き込み

入力パラメータ

AH=0EH

AL=文字コード

BL=属性

リターン情報

なし

機 能

カーソル位置へ文字コードと属性を書き込み、カーソルを進めます。

属性はグラフィック・モード時のみ有効です。

スクロール・アップした場合に生じるブランク行の属性は、次のようになります。

文字モード : 最下行の最初の桁の属性

グラフィック・モード : つねに 00H

また、次の4つの文字コードは制御コードとして扱われ、特殊な動作を行います。

コード	動 作
07H	ブザーを鳴らす。カーソルは移動しない
08H	カーソルを1文字分左へ移動
0AH	カーソルを1行下へ移動
0DH	カーソルを行の先頭へ移動

▶ INT10H(0FH) ビデオ・モードの取得	
入力パラメータ AH=0FH	リターン情報 AL = 現在のビデオ・モード AH = 1 行当たりの桁数 BH = 0 (ページ番号)
機 能 現在のビデオ情報を取得します。	

▶ INT10H(1000H) パレット・レジスタの設定	
入力パラメータ AX=1000H BH=カラー・レジスタ BL=パレット番号	リターン情報 なし
機 能 パレット・レジスタの設定を行います。	

▶ INT10H(1001H) オーバースキャン・レジスタの設定	
入力パラメータ AX=1001H BH=カラー・レジスタ	リターン情報 なし
機 能 オーバースキャン・レジスタの設定を行います。 オーバースキャン・レジスタは表示外枠の表示色を決定します。	

▶ INT10H(1002H) パレット・レジスタの一括設定	
入力パラメータ AX=1002H ES:DX=テーブル・アドレス	リターン情報 なし
機 能 パレット・レジスタの設定を一括に行います。 テーブルの内容は次のようになっています。 バイト 0 ~ 15 : パレット・レジスタへの設定値 バイト 16 : オーバースキャン・レジスタへの設定値	

▶ INT10H(1007H) パレット・レジスタの読み取り	
入力パラメータ AX=1007H BL=パレット番号	リターン情報 BH=カラー・レジスタ
機 能 パレット・レジスタの取得を行います。	

▶ INT10H(1008H) オーバースキャン・レジスタの読み取り	
入力パラメータ AX=1008H	リターン情報 BH=カラー・レジスタ
機 能 オーバースキャン・レジスタの取得を行います。 オーバースキャン・レジスタは表示外枠の表示色です。	

▶ INT10H(1009H) パレット・レジスタの一括読み取り	
入力パラメータ AX=1009H ES:DX=テーブル・アドレス	リターン情報 なし
機 能 パレット・レジスタの読み取りを一括に行います。 テーブルの内容は次のようになっています。 バイト 0～15: パレット・レジスタの値 バイト 16 : オーバースキャン・レジスタの値	

▶ INT10H(1010H) カラー・レジスタの設定	
入力パラメータ AX=1010H BX=カラー・レジスタ DH=設定する赤の輝度 CH=設定する緑の輝度 CL=設定する青の輝度	リターン情報 なし
機 能 カラー・レジスタの設定を行います。	

▶ INT10H(1012H) カラー・レジスタの一括設定	
入力パラメータ AX=1012H BX=先頭のカラー・レジスタ CX=登録カラー・レジスタ数 ES:DX=テーブル・アドレス	リターン情報 なし
機 能 カラー・レジスタの設定を一括に行います。 設定するテーブルの内容は次のようになっています。 バイト0～2: 赤の輝度, 緑の輝度, 青の輝度 バイト3～5: . . .	

▶ INT10H(1013H) カラー・レジスタの一括設定	
入力パラメータ AX=1013H BX=先頭のカラー・レジスタ CX=登録カラー・レジスタ数 ES:DX=テーブル・アドレス	リターン情報 なし
機 能 カラー・レジスタの設定を一括に行います。 設定するテーブルの内容は次のようになっています。 バイト0～2: 赤の輝度, 緑の輝度, 青の輝度 バイト3～5: . . .	

▶ INT10H(1015H) カラー・レジスタの読み取り	
入力パラメータ AX=1015H BX=カラー・レジスタ	リターン情報 DH=設定する赤の輝度 CH=設定する緑の輝度 CL=設定する青の輝度
機 能 カラー・レジスタの読み取りを行います。	

▶ INT10H(1017H) カラー・レジスタの一括読み取り	
入力パラメータ AX=1017H BX=先頭のカラー・レジスタ CX=取得カラー・レジスタ数 ES:DX=テーブル・アドレス	リターン情報 ES:DX=テーブル・アドレス
機 能 カラー・レジスタの読み取りを一括に行います。 読み取られたテーブルの内容は次のようになっています。 バイト0～2:赤の輝度, 緑の輝度, 青の輝度 バイト3～5:・・・	

▶ INT10H(1100H) ユーザ定義の文字フォント登録	
入力パラメータ AX=1100H BH=文字当たりのバイト数 BL=00H(設定ブロック) CX=設定する文字数 DX=登録する最初の文字コード ES:BP=テーブル・アドレス	リターン情報 なし
機 能 ユーザ定義の1バイト系文字フォントを登録します。	

▶ INT10H(1118H) 高密度文字フォントへの切り替え (V-Text)	
入力パラメータ AX=1118H BL=00H(設定ブロック)	リターン情報 なし
機 能 文字表示に使用するフォントを, 標準フォントから高密度表示用のフォントへ切り替えます。行数は変わりますが桁数は変わりません。 この機能は, 「ビデオ・モードの設定」(INT10H, AH=00H) 直後に呼び出されなければなりません。また, 現行モードが「文字密度の変更」(INT10H, AH=12H, BL=39H)によって, 高密度表示に設定されている必要があります。 この機能は, V-Text 対応のドライバがインストールされていなければ利用できません。	

▶ INT10H (1130H) 画面の行数情報 (V-Text)	
入力パラメータ AX=1130H BH=01H	リターン情報 CX=文字の高さ DL=1 画面の行数-1 ES: BP=INT43H のベクタ
機 能 画面の行数を取得します。 この機能は、V-Text 対応のドライバがインストールされていなければ利用できません。	

▶ INT10H (1131H) 拡張モード・テーブルの読み取り

(V-Text)

入力パラメータ AX=1131H	リターン情報 CX=エントリ数 ES: BP=テーブル・アドレス
---------------------	--

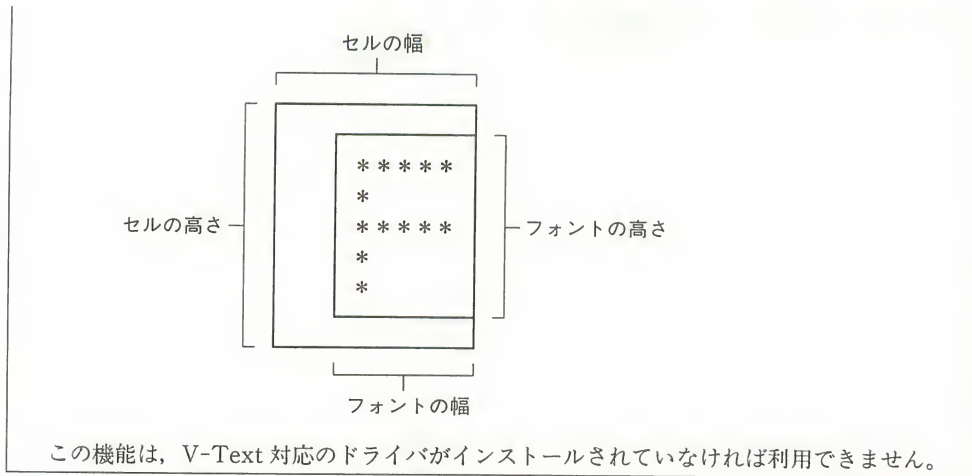
機能

V-Text モードのパラメータ・テーブルを取得します。

テーブルのそれぞれのエントリは 16 バイトで、次の構成を取ります。

オフセット	サイズ	内 容
00H	1 バイト	モード番号
01H	1 バイト	モード情報 ビット 7 : 使用不可のモード ビット 6 ~ 0 : 予約済み
02H	1 バイト	桁数
03H	1 バイト	行数
04H	1 バイト	文字セルの幅
05H	1 バイト	文字セルの高さ
06H	1 バイト	文字フォントの幅
07H	1 バイト	文字フォントの高さ
08H	1 バイト	予約済み(システムで使用)
0AH	1 ワード	予約済み(システムで使用)
0CH	1 ワード	予約済み
0EH	1 ワード	予約済み

セルとフォントの関係は次のようになっています。



▶ INT10H(12-10H) VGA 情報の取得(非公開)	
入力パラメータ AH=12H BL =10H	リターン情報 BH=0: カラー・モード 1: モノクロ・モード BL =VRAM サイズ 00H: 64K バイト 01H: 128K バイト 02H: 192K バイト 03H: 256K バイト CH =FEATURE 情報 CL =設定スイッチの値
機 能 VGA の設定情報を取得します。	

▶ INT10H(12-20H) プリント・スクリーン処理の切り替え	
入力パラメータ AH=12H BL =20H	リターン情報 なし
機 能 プリント・スクリーン処理(INT05H)を、ビデオ・カードの BIOS の処理に切り替えます。	

▶ INT10H(12-31H) モード変更時のパレット・ロード設定	
入力パラメータ AH=12H BL=31H AL=パレット・ロード設定 00H: 許可する 01H: 禁止する	リターン情報 AL=12H(機能サポート時)
機 能 ビデオ・モード設定時にデフォルトのパレットをロードするかどうかを設定します。	

▶ INT10H(12-32H) ビデオ・メモリへのアクセス設定	
入力パラメータ AH=12H BL=32H AL=アクセス設定 00H: 許可する 01H: 禁止する	リターン情報 AL=12H(機能サポート時)
機 能 ビデオ・メモリへの I/O アクセスを設定します。	

▶ INT10H(12-33H) グレースケールの設定	
入力パラメータ AH=12H BL=33H AL=グレースケール設定 00H: 許可する 01H: 禁止する	リターン情報 AL=12H(機能サポート時)
機 能 パレットのグレースケールへの変換を設定します。	

▶ INT10H(12-34H) カーソル・エミュレーションの設定	
入力パラメータ AH=12H BL=34H AL=エミュレーション設定 00H:許可する 01H:禁止する	リターン情報 AL=12H(機能サポート時)
機 能 文字の高さにより、カーソル形状の設定の計算方式を変更するかどうかの設定を行います。	

▶ INT10H(12-36H) ビデオ・スクリーンの ON/OFF	
入力パラメータ AH=12H BL=36H AL=スイッチ設定 00H:ON 01H:OFF	リターン情報 AL=12H(機能サポート時)
機 能 ビデオ・スクリーンの ON/OFF を制御します。	

▶ INT10H(12-38H) 文字フォント・サイズの変更 (V-Text)	
入力パラメータ AH=12H BL=38H AL=テーブル・インデックス BH=モード番号(AL=FFH)	リターン情報 AL=12H(機能サポート時)
機 能 ビデオ・モードごとの文字フォントのサイズを変更します。ただし、変更できるのはビデオ・モードの 03H と 73H のみです。 この機能により設定された文字フォントのサイズは、次回の「ビデオ・モードの設定」(INT10H, AH=00H)から有効になります。 AL のテーブル・インデックスは、「拡張モード・テーブルの読み取り」(INT10H, AX=1131H)で取得されたパラメータ・テーブルのエントリ番号で、-1(FFH)が指定されたときは、BH で指定されたビデオ・モードのフォント・サイズを標準設定に戻します。 この機能は、V-Text 対応のドライバがインストールされていなければ利用できません。	

▶ INT10H(12-39H) 表示文字密度の変更 (V-Text)	
入力パラメータ AH=12H BL=39H AL=テーブル・インデックス BH=モード番号(AL=FFH)	リターン情報 AL=12H(機能サポート時)
機 能 ビデオ・モードごとの表示文字密度を変更します。 この設定は、ビデオ・モード 70H と 71H の場合は「ビデオ・モードの設定」(INT10H, AH=00H), 03H と 73H の場合は「高密度文字フォントへの切り替え」(INT10H, AX=1118H)で有効となります。 AL のテーブル・インデックスは、「拡張モード・テーブルの読み取り」(INT10H, AX=1131H)で取得されたパラメータ・テーブルのエントリ番号で、-1(FFH)が指定されたときは、BH で指定されたビデオ・モードのフォント・サイズを標準設定に戻します。 この機能は、V-Text 対応のドライバがインストールされていなければ利用できません。	

▶ INT10H(12-3AH) 拡張ビデオ・モード設定情報の取得 (V-Text)	
入力パラメータ AH=12H BL=3AH AL=ビデオ・モード番号	リターン情報 AL=12H(機能サポート時) CH=文字フォント・サイズの設定 CL=文字密度の設定
機 能 拡張ビデオ・モードの設定情報を取得します。 この機能は、V-Text 対応のドライバがインストールされていなければ利用できません。	

▶ INT10H(1300H・01H・02H・03H) 文字列の書き込み	
入力パラメータ AH=13H AL=属性とカーソル移動制御 BH=00H(ページ番号) BL=属性(AL=0, 1 のとき) CX=書き込む文字列長 DH=書き込む行位置 DL=書き込む桁位置 ES:BP=文字列のアドレス	リターン情報 なし

機 能

指定位置へ文字列を書き込みます。

AL の値により、次の 4 つのモードが設定されています。

AL	カーソル	文字列の構成
00H	移動なし	文字コード, 文字コード...
01H	移動あり	同上
02H	移動なし	文字コード, 属性, 文字コード, 属性...
03H	移動あり	同上

また、文字列中の次の 4 つの文字コードは制御文字として扱われます。

コード	動 作
07H	ブザーを鳴らす。カーソルは移動しない
08H	カーソルを 1 文字分左へ移動
0AH	カーソルを 1 行下へ移動
0DH	カーソルを行の先頭へ移動

▶ INT10H(1310H・11H) 文字ブロックの読み取り
入力パラメータ

AH=13H

AL=属性モード

BH=00H(ページ番号)

CX=読み取る文字列長

DH=読み取る行位置

DL=読み取る桁位置

ES:BP=バッファ・アドレス

リターン情報

なし

機 能

指定位置から文字列を読み込みます。カーソルは移動しません。

AL の値により、次の 2 つのモードが設定されています。

AL	文字列の構成
10H	文字コード, 属性...
11H	文字コード, 属性 0, 属性 1, 属性 2... (ビデオ・モード 73H の場合のみ有効)

▶ INT10H(1320H・21H) 文字ブロックの書き込み

入力パラメータ

AH=13H

AL=属性モード

BH=00H(ページ番号)

CX=書き込む文字列長

DH=書き込む行位置

DL=書き込む桁位置

ES:BP=バッファ・アドレス

リターン情報

なし

機 能

指定位置に文字列を書き込みます。カーソルは移動しません。

ALの値により、次の2つのモードが設定されています。

AL	文字列の構成
20H	文字コード, 属性...
21H	文字コード, 属性0, 属性1, 属性2... (ビデオ・モード73Hの場合のみ有効)

▶ INT10H(1800H・01H) フォント・パターンの読み書き

入力パラメータ

AH=18H

AL=読み書きモード

00H:読み取り

01H:書き込み

BX=00H(文字セット)

CX=文字コード

DH=文字の横ドット数

DL=文字の縦ドット数

ES:SI=バッファ・アドレス

リターン情報

AL=00H:正常終了

それ以外:エラー

機 能

フォント・パターンの読み書きを行います。

書き込みができるのは、ユーザ定義文字のみです。

データの並びは、次のようになっています。

・8×16 または 8×19 ドットフォント(1 バイト文字セット)

0	0
1	1
⋮	⋮
15	18

・16×16 ドット・フォント(2 バイト文字セット)

0	1
2	3
⋮	⋮
30	31

・24×24 ドット・フォント(2 バイト文字セット)

0	1	2
3	4	5
⋮	⋮	⋮
69	70	71

なお、この機能は内部的にフォントの読み取り (INT15H, AH=50H) を呼び出しています。より高速な動作が必要な場合は、そちらを参照してください。

▶ INT10H (1A00H) ディスプレイ組み合わせコードの読み取り

入力パラメータ

AX=1A00H

リターン情報

AL=1AH (機能サポート時)

BL=ディスプレイ・コード

00H: ディスプレイなし

07H: VGA モノクロ

08H: VGA カラー

機 能

ディスプレイの組み合わせコードを読み取ります。

▶ INT10H(1DH) キーボード・シフト標識域の制御	
入力パラメータ AH=1DH AL = 標識域の制御 00H : 表示 01H : 消去 02H : 状況取得 BX = 標識域の行数 (AL=0)	リターン情報 BX = 標識域の行数 (AL=2)
機 能 キーボード・シフト標識域の制御を行います。 この機能は、入力支援サブシステム (\$IAS.SYS) のためのものなので、状況の取得 (AL=02H) を除いては、アプリケーションはこの機能を使用してはなりません。 表示の制御は、シフト状況の制御 (INT16H, AH=14H) を利用してください。	

▶ INT10H(FEH) ビデオ・バッファ・アドレスの読み取り	
入力パラメータ AH=FEH ES : SI = B800 : 0000H	リターン情報 ES : SI = ビデオ・バッファ
機 能 ビデオ・バッファのアドレスを読み取ります。 最初に ES : SI へ B800 : 0000H を設定しておくと、ハードウェア・ビデオ・バッファが存在する場合はアドレス値は不変となり、存在しない場合は疑似ビデオ・バッファのアドレスに変わります。 この機能は、ビデオ・モード 03H でのみ使用可能です。	

▶ INT10H(FFH) 画面表示の更新	
入力パラメータ AH=FFH CX = 更新する文字数 ES : SI = ビデオ・バッファ	リターン情報 なし
機 能 ビデオ・バッファの内容を更新します。 この機能は、ビデオ・モード 03H でのみ使用可能です。	

4.3 装置構成情報(INT11H)

このシステム構成情報は、BIOS ワークエリアの 0040:0010H の 1 ワードの値を返しています。

▶ INT11H 機器構成情報の読み取り	
入力パラメータ なし	リターン情報 AX=機器構成情報
機 能 機器構成情報を読み取ります。 AX のビット構成は次のようになっています。	
ビット 15 ~ 14	プリンタ・ポート数
ビット 13	予約済み
ビット 12	予約済み
ビット 11 ~ 9	RS-232SC ポート数
ビット 8	予約済み
ビット 7 ~ 6	接続 FDD 数-1
ビット 5 ~ 4	予約済み
ビット 3	未使用
ビット 2	マウスの有無
ビット 1	数値演算コプロセッサの有無
ビット 0	予約済み(つねに 1)

4.4 メモリサイズを得る(INT12H)

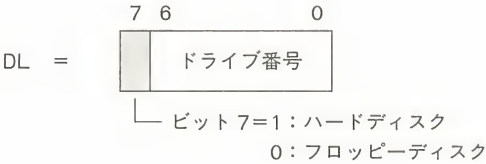
このメモリサイズは、BIOS ワークエリアの 0040:0013H の値を返しています。

▶ INT12H メモリサイズの取得	
入力パラメータ なし	リターン情報 AX=メモリサイズ
機 能 使用可能なメモリサイズを取得します。 単位は 1K バイト (1024 バイト) 単位で表されます。	

4.5 ディスク BIOS(INT13H)

ディスク BIOS は細かなディスクの制御が可能ですが、DOS/V の性質上、ディスクの操作はファンクション・コール (INT21H)で行うべきで、アプリケーションはこの BIOS を利用するべきではありません。

ディスク BIOS では、ドライブ番号の第 7 ビット目でフロッピーディスクとハードディスクを区別しています。



またハードディスクの場合は、シリンダ番号が 8 ビットでは表現できないために、次のように分割して格納しています。

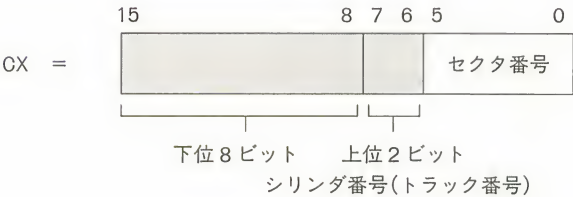


表 4-10 ディスク BIOS(INT13H)機能一覧

機能番号	機能内容
00H	ディスク・システムのリセット
01H	ディスク状況の読み取り
02H	セクタの読み込み
03H	セクタの書き込み
04H	セクタの検査
05H	シリンダ/トラックのフォーマット
08H	ディスクの情報の読み取り
09H	ハードディスク・ドライブの初期化
0CH	ハードディスクの SEEK
0DH	代替ハードディスク・リセット
11H	ハードディスクのヘッドの位置合わせ

15H	ドライブ・タイプの読み取り
16H	フロッピーディスクの入れ替え状況
18H	メディア・タイプの設定

以下に、ディスク BIOS (INT13H) の機能を各番号別に示します。

▶ INT13H(00H) ディスク・システムのリセット	
入力パラメータ AH=00H DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 ディスク・システムをリセットします。	

表 4-11 ディスク BIOS のエラー・ステータス

ステータス	内 容
00H	正常終了
01H	無効なディスク・パラメータが指定された
02H	アドレス・マークが見つからない
03H	書き込み禁止ディスクへの書き込みが行われた
04H	要求されたセクタ番号が見つからない
05H	リセットに失敗した
06H	ディスクの入れ替えが行われた
07H	ドライブ・パラメータ・アクティビティが失敗
08H	DMA オーバーラン
09H	64K バイト境界にまたがる DMA アクセスが発生
0AH	不良セクタ・フラグが検出された
0BH	不良シリングが検出された
0CH	指定されたメディア・タイプが見つからない
0DH	フォーマット時の無効なセクタ数
0EH	制御データ・アドレス・マーク検出
0FH	DMA のアービトレーション・レベルが範囲外
10H	読み込み時の CRC エラー
11H	ECC 訂正データ・エラー
12H	コマンドの処理中

13H	ドライブの電源が入っていない
20H	制御装置の障害
40H	SEEK 操作の障害
80H	フロッピーディスク・ドライブ作動不可
AAH	ハードディスク・ドライブ作動不可
BBH	確定できないエラーが発生した
CCH	書き込み時のエラー
EOH	状況エラー
FFH	Sense 操作の失敗

▶ INT13H(01H) ディスク状況の読み取り

入力パラメータ AH=01H DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 最後にディスクに対して実行された命令の状況を読み取ります。 エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H(02H) セクタの読み込み

入力パラメータ AH=02H AL = セクタ数 CX = シリンダ・セクタ番号 DH = ヘッド番号 DL = ドライブ番号 ES : BX = バッファ・アドレス	リターン情報 CF=0 の場合 正常終了 AL=読み込んだセクタ数 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 ディスクからセクタ単位でデータを読み込みます。 エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H(03H) セクタの書き込み	
入力パラメータ AH=03H AL=セクタ数 CX=シリンダ・セクタ番号 DH=ヘッド番号 DL=ドライブ番号 ES:BX=バッファ・アドレス	リターン情報 CF=0 の場合 正常終了 AL=書き込んだセクタ数 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 データをセクタ単位でディスクに書き込みます。 エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H(04H) セクタの検査	
入力パラメータ AH=04H AL=セクタ数 CX=シリンダ・セクタ番号 DH=ヘッド番号 DL=ドライブ番号 ES:BX=バッファ・アドレス	リターン情報 CF=0 の場合 正常終了 AL=読み込んだセクタ数 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 ディスクからセクタ単位でデータを読み込み、次の条件を検査します。 <ol style="list-style-type: none"> 1. セクタが存在するか 2. セクタは読み込めるか 3. データの CRC が正しいか エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H(05H) シリンダ/トラックのフォーマット

入力パラメータ

AH=05H

AL=セクタ数

CX=シリンダ番号

DH=ヘッド番号

DL=ドライブ番号

ES:BX=バッファ・アドレス

リターン情報

CF=0 の場合

正常終了

CF=1 の場合

エラー終了

AH=エラー・ステータス

機能

ディスクのフォーマットを行います。バッファの内容はフロッピーディスクとハードディスクで異なり、次のように指定します。

・フロッピーディスクの場合

オフセット	内 容
0	トラック番号
1	ヘッド番号
2	セクタ番号
3	セクタ当たりのバイト数 00H= 128K バイト/セクタ 01H= 256K バイト/セクタ 02H= 512K バイト/セクタ 03H= 1024K バイト/セクタ

なお、フォーマットの実行の前に、ディスク・タイプとメディア・タイプの設定を行っておく必要があります。

・ハードディスクの場合

次のデータをシリンダ当たりのセクタ数分だけ準備する必要があります。

オフセット	内 容
0	選択フラグ=00H:良セクタ 80H:不良セクタ
1	セクタ数

エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。

▶ INT13H(08H) ディスクの情報の読み取り	
入力パラメータ AH=08H DL = ドライブ番号	リターン情報 CX = 最大シリンダ・セクタ数 DH = 最大ヘッド番号 DL = 登載ドライブ数 ・フロッピーディスクの場合のみ AX = 000H BH = 00H BL = ドライブ・タイプ ES:DI = テーブル・アドレス
機 能 ディスクの情報を読み取ります。 ドライブがフロッピーディスクの場合のみ、次の情報が取得されます。 ・BL=ドライブ・タイプ 01H: 2D 02H: 2HD(1.2M バイト) 03H: 2DD(720K バイト) 04H: 2HD(1.44M バイト) 06H: 2ED(2.88M バイト) ・ES:DI=メディア・パラメータ・テーブル(11 バイト) ドライブがサポートしている最大のメディア・タイプのパラメータ・テーブル。 エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H(09H) ハードディスク・ドライブの初期化	
入力パラメータ AH=09H DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 ハードディスク・ドライブの初期化を行います。	

▶ INT13H(0CH) ハードディスクの SEEK	
入力パラメータ AH=0CH CX = シリンダ番号 DH = ヘッド番号 DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 ハードディスクのヘッドを指定されたシリンダへ移動させます。 エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H(0DH) 代替ハードディスク・リセット	
入力パラメータ AH=0DH DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 代替ハードディスクをリセットします。	

▶ INT13H(11H) ハードディスクのヘッドの位置合わせ	
入力パラメータ AH=11H DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 ハードディスクのヘッドをシリンダ0の位置に戻します。 エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H(15H) ドライブ・タイプの読み取り	
入力パラメータ AH=15H DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 AH=ドライブのタイプ CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 ドライブのタイプを読み取ります。 ・AH=ドライブのタイプ 00H：ドライブが接続されていない 01H：フロッピーディスクの入れ替えを検知できない 02H：フロッピーディスクの入れ替えを検知できる 03H：ハードディスク エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H (16H) フロッピーディスクの入れ替え状況	
入力パラメータ AH=16H DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 AH=ディスクの入れ替え状況 CF=1 の場合 エラー終了 AH=エラー・ステータス
機 能 フロッピーディスクの入れ替え状況を読み取ります。 ディスクの入れ替え状況は次の意味をもちます。 ・AH=ディスクの入れ替え状況 00H：ディスクの入れ替えを検知していない 01H：無効なディスク・パラメータ 06H：ディスクの入れ替えを検知した 80H：ディスク・ドライブ作動不可 エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

▶ INT13H (18H) メディア・タイプの設定	
入力パラメータ AH=18H CX = シリンド・セクタ数 DL = ドライブ番号	リターン情報 CF=0 の場合 正常終了 ES:DI=テーブルアドレス CF=1 の場合 エラー終了 AL=エラー・ステータス
機 能 ディスクのメディア・タイプを設定します。 フォーマット時やディスクが入れ替えられた際にも、この機能で設定を行う必要があります。 リターン情報として、ES:DI にそのメディアのパラメータ・テーブルの情報が取得できます。 エラー終了した場合は、ディスク・システムのリセットを実行してから、再度命令をやりなおしてください。	

※ディスク・ドライブのパラメータ・テーブル

ディスク BIOS が参照するディスク・ドライブのパラメータ・テーブルは次の割り込みベクタで示されています。

INT 1EH : フロッピー・ドライブ
 INT 41H : ハードディスク・ドライブ 0
 INT 46H : ハードディスク・ドライブ 0

それぞれの内容は、次のようになっています。

表 4-12 フロッピーディスク・ドライブのパラメータ・テーブル

オフセット	データ長	内 容
00H	1 バイト	ステップ・レイト, ヘッド・アンロード・タイム
01H	1 バイト	DMA モードにおけるヘッド・ロード・タイム
02H	1 バイト	ドライブの回転停止までの待ち時間 (55ms)
03H	1 バイト	セクタ当たりのバイト数 00H: 128 バイト/セクタ 01H: 256 バイト/セクタ 02H: 512 バイト/セクタ 03H: 1024 バイト/セクタ
04H	1 バイト	トラック当たりのセクタ数
05H	1 バイト	ギャップの長さ
06H	1 バイト	データの長さ
07H	1 バイト	フォーマットに関するギャップの長さ
08H	1 バイト	フォーマットに関するバイトの数
09H	1 バイト	ヘッドの安定時間 (1ms)
0AH	1 バイト	モーターの起動時間 (1/8 秒)

表 4-13 ハードディスク・ドライブのパラメータ・テーブル

オフセット	データ長	内 容
00H	1 ワード	最大シリンダ数
02H	1 バイト	最大ヘッド数
03H	1 ワード	未使用
05H	1 ワード	代替シリンダ書き込み開始
07H	1 バイト	未使用
08H	1 バイト	制御バイト ビット 7 ~ 6 : 再試行不可 ビット 5 : 出荷時の不良シリンダ表示 ビット 3 : ヘッドが 8 以上
09H	3 バイト	未使用
0CH	1 ワード	ランディング領域
0EH	1 バイト	トラック当たりのセクタ数
0FH	1 バイト	予約済み

4.6 RS-232C・BIOS(INT14H)

RS-232C・BIOS では、最大 4 チャンネルまでのシリアル・ポートを管理できます。

しかし、この BIOS は受信が割り込み駆動でないため、高速な通信を行った場合などはデータを取りこぼすことも多く、現実的には直接ハードウェアを制御せざるをえない状況にあります。

この場合でも、IBM-PC では RS-232C のポート・アドレスは厳密には規定されていないので、必ず BIOS ワークエリアを参照して、動的な割り付けを行うようにしてください。

表 4-14 RS-232C・BIOS(INT14H)機能一覧

機能番号	機能内容
00H	通信ポートの初期設定
01H	文字の送信
02H	文字の受信
03H	通信ポート状況

以下に、RS-232C・BIOS(INT14H)の機能を各番号別に示します。

▶ INT14H(00H) 通信ポートの初期設定	
入力パラメータ AH=00H AL=設定値 DX=論理ポート番号	リターン情報 AH=回線制御状況 AL=モデム状況
機 能 通信ポートを初期化します。 設定値は次の意味をもちます。 ・AL=設定値 ビット 7～5：ボーレート 000 = 110bps 001 = 150bps 010 = 300bps 011 = 600bps 100 = 1200bps 101 = 2400bps 110 = 4800bps	

111 = 9600bps
 ビット 4 ~ 3 : パリティ
 00 = なし
 01 = 奇数
 11 = 偶数
 ビット 2 : ストップ・ビット
 0 = 1 ビット
 1 = 2 ビット
 ビット 1 ~ 0 : データ長
 10 = 7 ビット
 11 = 8 ビット

表 4-15 RS-232C・BIOS の回線制御状況ステータス

ビット	内 容
7	タイムアウト
6	送信シフト・レジスタが空
5	送信用保持レジスタが空
4	ブレーク信号を検出
3	フレーミング・エラー
2	パリティ・エラー
1	オーバーラン・エラー
0	受信データあり

表 4-16 RS-232C・BIOS のモデム状況ステータス

ビット	内 容
7	受信回線信号の検出(CD)
6	呼び出し信号(RI)受信
5	データ・セット・レディ(DSR)
4	送信可(CTS)
3	受信回線信号の検出に変化あり
2	呼び出し信号(RI)受信の終端検出
1	データ・セット・レディに変化あり
0	送信可に変化あり

▶ INT14H(01H) 文字の送信	
入力パラメータ AH=01H AL=送信する文字 DX=論理ポート番号	リターン情報 AH=回線制御状況 AL=送信した文字
機 能 1文字送信します。このときはDTR信号とRTS信号はONになります。 データが送信できなかったとき(規定時間以内にDSR信号とCTS信号がONにならなかったときは、タイムアウトで終了します。	

▶ INT14H(02H) 文字の受信	
入力パラメータ AH=02H DX=論理ポート番号	リターン情報 AH=回線制御状況 AL=受信された文字
機 能 1文字受信します。 受信が完了するか、タイムアウトになるまで待ち状態になります。	

▶ INT14H(03H) 通信ポート状況	
入力パラメータ AH=03H DX=論理ポート番号	リターン情報 AH=回線制御状況 AL=モデム状況
機 能 通信ポートの状況を読み取ります。	

4.7 システム・サービス BIOS(INT15H)

システム・サービス BIOS は、デバイス関連以外の各種サービスを集めた BIOS ですが、DOS/V の数ある BIOS の中でも、最も扱いの難しい BIOS でもあります。

その理由は、この BIOS で対応している機能が AT 互換機と PS/2 互換機で異なっており、互換性維持のために DOS/V では大幅な制限を加えているからなのです。

OADG の規定でも、この BIOS はできる限り使わずに、別の機能で代用するように指示されています。IBM-PC 関連の資料をみると、本書にはない多くの機能が取り上げられているかもしれませんが、これらは DOS/V 全体の互換性を考慮したうえで削除したもののなのです。注意してください。

表 4-17 システム・サービス BIOS の機能一覧

機能番号	機能内容
49H	BIOS タイプの取得
4FH	キーボード・インターセプト
50H	フォントの読み書き機能のアドレス取得
87H	メモリ・ブロックの移動
88H	拡張メモリ・サイズの取得

以下に、システム・サービス BIOS (INT15H) の機能を各機能番号別に示します。

▶ INT15H(49H) BIOS タイプの取得	
入力パラメータ AH=49H AL=00H	リターン情報 CF=0 の場合 AH=00H(機能サポート) BL=BIOS タイプ CF=1 の場合 AH=86H(サポートなし)
機 能 現行の BIOS タイプを読み取ります。 この機能により、現在使用中の DOS が DOS/V であるかどうかの判定が可能です。ただし、この判定は、この機能がサポートされていない機種上で行われる可能性があるもので、機能サポートが行われていることを必ず確認しなければなりません。 BIOS タイプは次のように設定が行われています。	

・BL=BIOS タイプ

00H : DOS/V・BIOS, または PS/2・BIOS

01H : DOS・J4.0 以上の BIOS

その他: 予約済み

▶ INT15H(4FH) キーボード・インターセプト

入力パラメータ

AH=4FH

AL=走査コード

リターン情報

CF=1 の場合

AL=新しい走査コード

CF=1 の場合

走査コードには変化なし

機 能

キーボードの走査コードの変換を行います。

この機能は、キーボードのハードウェア割り込み(INT09H)から呼び出され、必要があれば走査コードを変換して戻ります。

したがって、これと同様な機能のプログラムを作成し、チェーンすれば、独自にキーの配置を変更することも可能です。

▶ INT15H(50H) フォントの読み書き機能のアドレス取得

入力パラメータ

AH=50H

AL=取得モード

00H: 読み込み機能

01H: 書き込み機能

BH=フォントの種類

ビット 0=0:1 バイト文字セット

1:2 バイト文字セット

BL=0(予約済み)

DH=フォントの横ドット数

DL=フォントの縦ドット数

BP=0(コードページ)

リターン情報

CF=0 の場合

正常終了

ES:BX=機能アドレス

CF=1 の場合

エラー終了

AH=エラー・ステータス

機 能

フォントの読み込み・書き込み機能のアドレスを取得します。

CX に文字コードを設定して、取得されたアドレスを far コールすることで ES:SI にフォント・イメージが取得されます。

ビデオ BIOS のフォント取得も、この機能呼び出しているの、高速に処理を行いたい場合は、この機能を利用するとよいでしょう。

・エラー・ステータス

ステータス	内 容
00H	正常終了
01H	フォントの種類(BH)が無効
02H	BLの設定が無効
03H	フォント・サイズ(DH, DL)が無効
04H	コード・ページ(BP)が無効
86H	機能はサポートされていない

8×16 ドット

8 ドット

0
1
⋮
15

8×19 ドット

8 ドット

0
1
⋮
18

16×16 ドット

8 ドット 8 ドット

0	1
2	3
⋮	⋮
14	15

12×24 ドット

8 ドット 4 ドット

0	1	
2	3	
⋮	⋮	
46	47	

24×24 ドット

8 ドット 8 ドット 8 ドット

0	1	2
3	4	5
⋮	⋮	⋮
69	70	71

図 4-1

フォント・パターンのデータ構造(バイト・オフセット順)

List 4-1

高速なフォントの読み込みプログラム

```

GET_FGETADR:                                読み込みアドレス取得
      MOV     AX, 5000H
      MOV     BH, 00H
      MOV     BL, 00H
      MOV     DH, 08H
      MOV     DL, 10H
      MOV     BP, 0
      INT     15H
      MOV     WORD PTR [FGETADR], BX
      MOV     WORD PTR [FGETADR+2], ES
      RET

;
; CX = 文字コード (BUFFER にボタンが取得される)
;

GET_FONT:                                    ; フォント読み込み
      PUSH    AX

      MOV     AX, CS
      MOV     ES, AX
      LEA     SI, BUFFER
      CALL    [FGETADR]

      POP     AX
      RET

FGETADR      DD      ?                      ; 読み込みアドレス

BUFFER       DB      16 DUP (?)           ; フォントバッファ

```

▶ INT15H(87H) メモリ・ブロックの移動

入力パラメータ

AH=87H

CX = 転送ワード数

ES : SI=GDT アドレス

リターン情報

CF=0 の場合

正常終了

CF=1 の場合

AH=エラー・ステータス

機 能

1M バイト以降のプロテクト・メモリ間とのデータ転送を行います。

GDT(グローバル記述子テーブル)は次の構成を取ります。

オフセット	サイズ	内 容
00H	8 バイト	ダミー (0 で初期化)
08H	8 バイト	GDT の位置 (0 で初期化)
10H	1 ワード	転送元セグメント長
	3 バイト	24 ビット・アドレス
	1 バイト	セグメントアクセス権 (93H)
	1 ワード	未使用
18H	1 ワード	転送先セグメント長
	3 バイト	24 ビット・アドレス
	1 バイト	セグメントアクセス権 (93H)
	1 ワード	未使用
20H	8 バイト	BIOS・CS (0 で初期化)
28H	8 バイト	SS (0 で初期化)

・ エラー・ステータス

ステータス	内 容
00H	正常終了
01H	RAM パリティ・エラー
02H	他の例外割り込みエラー
03H	ゲート・アドレス・ライン 20H 失敗

▶ INT15H(88H) 拡張メモリ・サイズの取得

入力パラメータ

AH=88H

リターン情報

AX=1M バイト以降のサイズ

機 能

1M バイト以降のプロテクト・メモリ・サイズを取得します。

単位は 1K バイト (1024 バイト) です。

DOS/V では \$FONT.SYS がこの機能をフックすることで、フォント・ロード用の領域を確保しています。

4.8 キーボード BIOS(INT16H)

キーボード BIOS は、多機種キーボードの対応、2 バイト文字系への対応など課題の多い BIOS です。

特に多機種キーボードへの対応は非常に困難なので、注意を要します。このあたりの詳細な話題は第 2 章でまとめてありますから、ここでは一般的な BIOS の紹介にとどめておきます。

IBM-PC のキーボード BIOS では、キーが押されるとそのキー固有の番号である「走査コード」と、そのキーに刻印してある「文字コード」を返します。このうち「走査コード」はキーボードごとに異なるため、これにより判定を行ってしまうと、別のキーボードで動作しないといった状況が生じかねません。

そこで OADG の規定でも、この走査コードは判断に利用せず、文字コードのみで判定するように提案されています。

OADG サポートのキーボードに関しては、その詳細を「Appendix」の A-2 ～ A-5 に示してあります。参考にしてください。

表 4-18 キーボード BIOS(INT16H)機能一覧

機能番号	機能内容
00H	1 文字入力待ち(84 キーボード)
01H	文字の入力状況(84 キーボード)
02H	キーボードのシフト状況(84 キーボード)
0305H	キーボード・タイプ速度の設定
05H	キー・バッファへの書き込み
10H	1 文字入力待ち(拡張キーボード)
11H	文字の入力状況(拡張キーボード)
12H	キーボードのシフト状況(拡張キーボード)
1300H	DBCS の状況モード設定
1301H	DBCS の状況モード取得
14H	シフト状況の制御

以下に、キーボード BIOS(INT16H)の機能を各番号別に示します。

▶ INT16H(00H) 1 文字入力待ち(84 キーボード)	
入力パラメータ AH=00H	リターン情報 AH=走査コード AL=文字コード
機 能 1 文字入力します。入力があるまでウェイトします。	

▶ INT16H(01H) 文字の入力状況(84 キーボード)	
入力パラメータ AH=01H	リターン情報 ZF=0 の場合 入力データあり AH=走査コード AL=文字コード ZF=1 の場合 入力データなし
機 能 文字の入力状況を調査します。入力がなくてもすぐに戻ってきます。 この機能ではキーボード・バッファの内容を調べるだけで、あらためて文字入力を行う までバッファは更新されません。	

▶ INT16H(02H) キーボードのシフト状況(84 キーボード)	
入力パラメータ AH=02H	リターン情報 AH=予約済み AL=シフト情報
機 能 キーボードのシフト状況を読み取ります。 シフト情報は次の意味をもち、押されているものにビットが立ちます。 ・AH=予約済み ・AL=シフト情報 ビット 7 : Insert キー ビット 6 : CapsLock キー ビット 5 : NumLock キー ビット 4 : ScrollLock キー ビット 3 : Alt キー ビット 2 : Ctrl キー ビット 1 : 左シフトキー ビット 0 : 右シフトキー	

▶ INT16H(0305H) キーボード・タイプ速度の設定

入力パラメータ

AX=0305H

BH=リピート待ち時間

BL=リピート間隔

リターン情報

なし

機能

キーボード・タイプ速度を設定します。

・BH=リピート待ち時間

00H : 250 ミリ秒

01H : 500 ミリ秒

02H : 750 ミリ秒

03H : 1000 ミリ秒

・BL=リピート間隔(1秒あたりにタイプできる文字数)

設定値	文字数
00H	30.0
01H	26.7
02H	24.0
03H	21.8
04H	20.0
05H	18.5
06H	17.1
07H	16.0
08H	15.0
09H	13.3
0AH	12.0

設定値	文字数
0BH	10.9
0CH	10.0
0DH	9.2
0EH	8.6
0FH	8.0
10H	7.5
11H	6.7
12H	6.0
13H	5.5
14H	5.0
15H	4.6

設定値	文字数
16H	4.3
17H	4.0
18H	3.7
19H	3.3
1AH	3.0
1BH	2.7
1CH	2.5
1DH	2.3
1EH	2.1
1FH	2.0

▶ INT16H(05H) キー・バッファへの書き込み

入力パラメータ

AH=05H

CH=走査コード

CL=文字コード

リターン情報

AL=エラー・ステータス

00H: 正常終了

01H: バッファがいっぱい

機能

キーボードから入力されたものと同様に、キー・バッファに走査コードと文字コードを書き込みます。

▶ INT16H(10H) 1文字入力待ち(拡張キーボード)	
入力パラメータ AH=10H	リターン情報 AH=走査コード AL=文字コード
機 能 1文字入力を待ちます。入力があるまでウェイトします。 読み取られるコードの一部に違いがある他は、INT16H(00H)と同様です。	

▶ INT16H(11H) 文字の入力状況(拡張キーボード)	
入力パラメータ AH=11H	リターン情報 ZF=0 の場合 入力データあり AH=走査コード AL=文字コード ZF=1 の場合 入力データなし
機 能 文字の入力状況を調査します。入力がなくてもすぐに戻ってきます。 この機能ではキーボード・バッファの内容を調べるだけで、あらためて文字入力を行うまでバッファは更新されません。 読み取られるコードの一部に違いがある他は、INT16H(00H)と同様です。	

▶ INT16H(12H) キーボードのシフト状況(拡張キーボード)	
入力パラメータ AH=12H	リターン情報 AH=拡張シフト情報 AL=シフト情報
機 能 キーボードのシフト状況を読み取ります。 シフト情報は次の意味をもち、押されているものにビットが立ちます。	
・ AH=拡張シフト情報 ビット 7 : SysRq キー ビット 6 : CapsLock キー ビット 5 : NumLock キー ビット 4 : ScrollLock キー	・ AL=シフト情報 ビット 7 : Insert キー ビット 6 : CapsLock キー ビット 5 : NumLock キー ビット 4 : ScrollLock キー

ビット 3 : 右 Alt キー
 ビット 2 : 右 Ctrl キー
 ビット 1 : 左 Alt キー
 ビット 0 : 左 Ctrl キー

ビット 3 : Alt キー
 ビット 2 : Ctrl キー
 ビット 1 : 左シフトキー
 ビット 0 : 右シフトキー

▶ INT16H(1300H) DBCS の状況モード設定

入力パラメータ

AX=1300H

DX=状況モード

リターン情報

なし

機 能

DBCS(2 バイト文字セット)の状況モードを設定します。

・DX=状況モード

ビット 15 ~ 8 : 予約済み
 ビット 7 : 漢字モード
 ビット 6 : ローマ字モード
 ビット 5 ~ 3 : 予約済み
 ビット 2 ~ 1 : 00:英数シフト
 01:カタカナ・シフト
 10:ひらがなシフト
 11:予約済み
 ビット 0 : 全角モード

この機能は、入力支援サブシステム(\$IAS.SYS)の機能です。
 システムが導入されていなければ利用できません。

▶ INT16H(1301H) DBCS の状況モード取得

入力パラメータ

AX=1301H

リターン情報

DX=状況モード

機 能

DBCS(2 バイト文字セット)の状況モードを取得します。

この機能は、入力支援サブシステム(\$IAS.SYS)の機能です。
 システムが導入されていなければ利用できません。

▶ INT16H(14H) シフト状況の制御	
入力パラメータ AH=14H AL=00H: シフト状況の表示 01H: シフト状況の消去 02H: 表示状態取得	リターン情報 AL=表示状態 00H: 表示 01H: 消去
機能 画面最下行のキーボード・シフト状況の表示を制御します。 この機能は、入力支援サブシステム(\$IAS.SYS)の機能です。 システムが導入されていない場合は利用できません。	

4.9 プリンタ BIOS(INT17H)

DOS/V では、プリンタの制御コードは、エプソン提唱の ESC/P J84 が規定されています。

この規格では、全角文字は JIS コードで扱っていますが、プリンタ BIOS ではシフト JIS コードから JIS コードへの変換を自動的にを行い、かつその他の制御コードは透過になるように工夫されています。

表 4-19 プリンタ BIOS(INT17H)機能一覧

機能番号	機能内容
00H	文字の印刷
01H	プリンタ・ポートの初期設定
02H	状況の読み取り

以下に、プリンタ BIOS(INT17H)の機能を各番号別に示します。

▶ INT17H(00H) 文字の印刷	
入力パラメータ AH=00H AL=印刷する文字 DX=論理プリンタ番号	リターン情報 AH=状況ステータス
機 能 1 バイトだけプリンタへ出力します。 全角文字の場合、第 1 バイト出力の後、ただちに第 2 バイトを出力するようにしてください。	

▶ INT17H(01H) プリンタ・ポートの初期設定	
入力パラメータ AH=01H DX=論理プリンタ番号	リターン情報 AH=状況ステータス
機 能 ハードウェアの初期設定、ソフトウェア状況のリセット、初期制御値の設定等を行います。	

▶ INT17H(02H) 状況の読み取り	
入力パラメータ AH=02H DX = 論理プリンタ番号	リターン情報 AH = 状況ステータス
機 能 状況のステータスを読み込みます。	

表 4-20 プリンタ BIOS の状況ステータス

ビット	内 容
7	プリンタ動作可能
6	要求した命令に対する応答
5	用紙切れ, または自動給紙機構中の用紙詰まり
4	オンライン
3	I/O エラー
2 ~ 1	予約済み
0	タイムアウト

4.10 タイマ・クロック BIOS(INT1AH)

タイマ・クロック BIOS は、システム・タイマとリアルタイム・クロック(時計)の制御を行います。

この BIOS では、システム・タイマとリアルタイム・クロックの両方を同時に設定する必要があるため、設定に関してはファンクション・コール(INT21H)を利用してください。

表 4-21 タイマ・クロック BIOS(INT1AH)機能一覧

機能番号	機能内容
00H	システム・タイマの時刻カウンタの取得
01H	システム・タイマの時刻カウンタの設定
02H	リアルタイム・クロックの時刻の取得
03H	リアルタイム・クロックの時刻の設定
04H	リアルタイム・クロックの日付の取得
05H	リアルタイム・クロックの日付の設定

以下に、タイマ・クロック BIOS(INT1AH)の機能を各番号別に示します。

▶ INT1AH(00H) システム・タイマの時刻カウンタの取得	
入力パラメータ AH=00H	リターン情報 CX:DX=タイマ値 AL=日更新フラグ 0:24時を経過していない
機 能 システム・タイマのカウント値を取得します。	

▶ INT1AH(01H) システム・タイマの時刻カウンタの設定	
入力パラメータ AH=01H CX:DX=タイマ値	リターン情報 なし
機 能 システム・タイマのカウント値を設定します。 カウントは、1193180/65536 回/秒(毎秒 18.2 回)の割合で増加します。	

▶ INT1AH(02H) リアルタイム・クロックの時刻の取得	
入力パラメータ AH=02H	リターン情報 CF=0 の場合 クロック動作中 CH=時(BCD) CL=分(BCD) DH=秒(BCD) DL=夏時間制 0: 夏時間制選択 1: 夏時間制なし CF=1 の場合 クロックが動作していない
機 能 リアルタイム・クロックの時刻を取得します。	

▶ INT1AH(03H) リアルタイム・クロックの時刻の設定	
入力パラメータ AH=03H CH=時(BCD) CL=分(BCD) DH=秒(BCD) DL=夏時間制 0: 夏時間制選択 1: 夏時間制なし	リターン情報 なし
機 能 リアルタイム・クロックの時刻を設定します。	

▶ INT1AH(04H) リアルタイム・クロックの日付の取得	
入力パラメータ AH=04H	リターン情報 CF=0 の場合 クロック動作中 CH=世紀(BCD) CL=年(BCD) DH=月(BCD) DL=日(BCD) CF=1 の場合 クロックが動作していない
機 能 リアルタイム・クロックの日付を取得します。	

▶ INT1AH(05H) リアルタイム・クロックの日付の設定	
入力パラメータ AH=05H CH=世紀(BCD) CL=年(BCD) DH=月(BCD) DL=日(BCD)	リターン情報 なし
機 能 リアルタイム・クロックの日付を設定します。	

第 5 章

マウス BIOS

マウス BIOS は、他の BIOS と異なり、組み込み型の BIOS です。

その理由は、IBM-PC ではマウスのハードウェア的实现方法が多岐にわたるために、その設定状況に応じて制御方法を変える必要があったためです。

このマウス BIOS は、基本的にはマイクロソフト社のマウス・ドライバの規格に準拠しています。

5.1 マウス BIOS の処理系

(1) ビデオ・モード

マウス BIOS の対応するビデオ・モードは、ビデオ BIOS の対応するビデオ・モードと同一です。DOS/V 日本語モードではテキスト・モードもグラフィックスで仮想的に実現している関係上、いつでもマウスが使える状態にあります。

座標系は以下のように設定されており、X 方向(横)はつねに 640 に固定ですが、縦はビデオ・モードにより異なります。

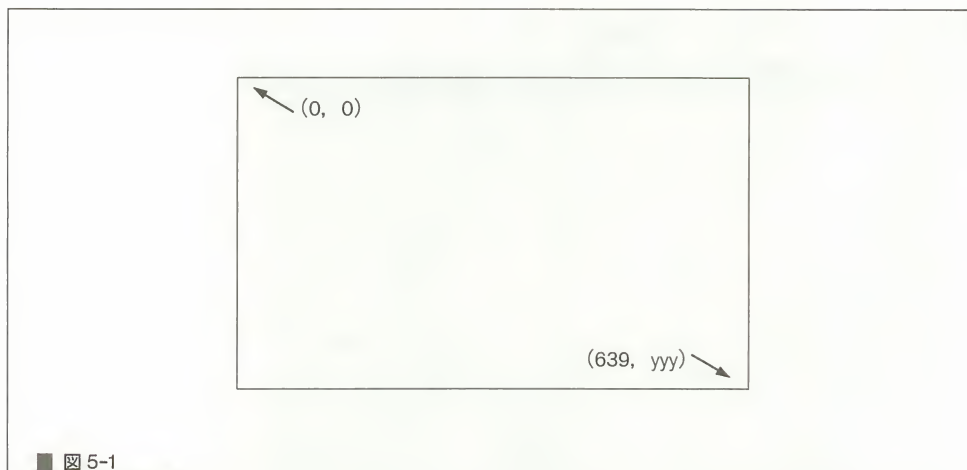


図 5-1

マウスの座標

表 5-1 各ビデオ・モードの解像度

モード	タイプ	解像度	色数
03H	文字モード	640×475 *	16
11H	グラフィック・モード	640×480	2
12H	グラフィック・モード	640×480	16
70H	V-Text 文字モード	可変	16
71H	V-Text 拡張文字モード	可変	16
72H	グラフィック・モード	640×480	16
73H	拡張文字モード	640×475 *	16

※ V-Text モードは、V-Text 対応ドライバがインストールされていない場合は利用できない

*がついている解像度は、V-Text 時は可変

(2) カーソル

カーソルはビデオ・モードにより表示の形式が変化します。

① グラフィック・カーソル

ビデオ・モードがグラフィック・モードの場合は、16×16 ドットの矢印カーソルが表示されます。このグラフィック・カーソルはパターンを変更することで形状を変化させることができます。

パターン定義には、AND マスクと XOR マスクと指示点情報が存在します。グラフィック・カーソルの表示は、この AND マスクでカーソル外形が消去された後、XOR マスクで形状が描画されます。

標準のグラフィック・カーソル(指示点(0, 0))

AND マスク	XOR マスク
0011111111111111	0000000000000000
0001111111111111	0100000000000000
0000111111111111	0110000000000000
0000011111111111	0111000000000000
0000001111111111	0111100000000000
0000000111111111	0111110000000000
0000000011111111	0111111000000000
0000000001111111	0111111100000000
0000000000111111	0111100000000000
0000000011111111	0110110000000000
0000000011111111	0100110000000000
0000000011111111	0000011000000000
0011000011111111	0000011000000000
1111000011111111	0000001100000000
1111100001111111	0000001100000000
1111100001111111	0000001100000000
1111100001111111	0000000000000000

② テキスト・カーソル

ビデオ・モードがテキスト・モードの場合は、テキスト・カーソルが表示されます。テキスト・カーソルは1文字単位で移動するカーソルで、基本的にはその位置での属性の反転を取ります。

(3) カーソル移動距離の単位

マウスの物理的な移動単位は、約 0.5mm を 1 として「マウス△」と表現します。画面上では、この「マウス△」当たりカーソルが何ドット移動するかを移動比率として表現し、設定することができます。

(4) マウスの動作確認

マウスは他のハードウェアと異なり、「接続されていない」という状態が普通に起こります。そればかりか、マウス・ドライバ自体がインストールされていないことも想定されます。

したがって、アプリケーションはこれらの状態を確認してから、マウスの機能を使用しなければなりません。

List 5-1

マウスの初期化

```

;
; マウスの使用可能検査(使用不可ならばZフラグが立つ)
;
MOUSE_INIT:                                ; ドライバ存在検査
      MOV     AH, 35H
      MOV     AL, 33H
      INT     21H
      MOV     AX, ES
      OR      AX, BX
      JZ      MOUSE_INIT_ER                ; ドライバ初期化

      MOV     AX, 0000H
      INT     33H
      OR      AX, AX
      JZ      MOUSE_INIT_ER                ; カーソル表示

      MOV     AX, 0001H
      INT     33H

      OR      AX, AX
MOUSE_INIT_ER:
      RET

```

5.2 マウス BIOS の機能

マウス BIOS の機能について表 5-2 に示します。

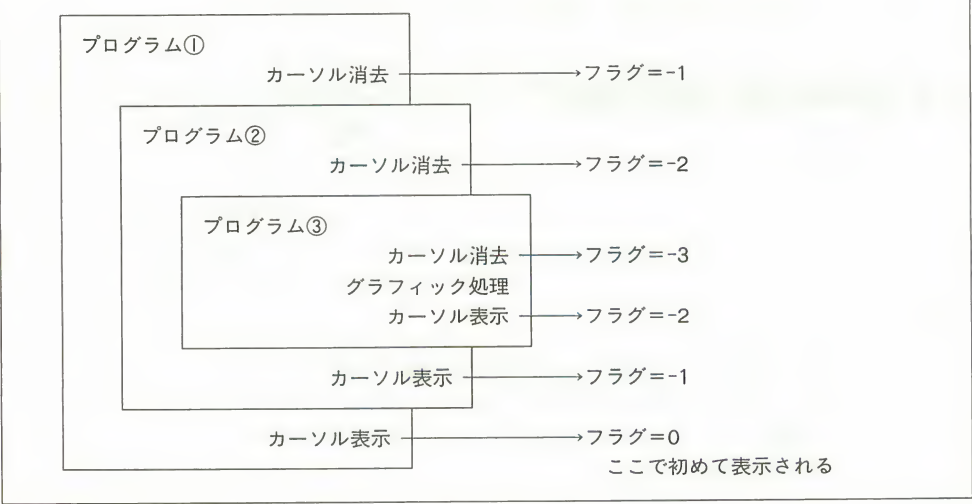
表 5-2 マウス BIOS(INT33H)機能一覧

機能番号	機能内容
0000H	マウスの初期化
0001H	カーソルの表示
0002H	カーソルの消去
0003H	カーソル位置とボタン情報の取得
0004H	カーソルの移動
0005H	ボタンが押された回数と最終位置の取得
0006H	ボタンが離された回数と最終位置の取得
0007H	カーソルの X 方向の移動範囲設定
0008H	カーソルの Y 方向の移動範囲設定
0009H	グラフィック・カーソルの形状設定
000AH	テキスト・カーソルの形状設定
000BH	カーソルの相対移動距離の取得
000CH	割り込みサブルーチンの設定
000DH	ライトペン・エミュレーションの ON
000EH	ライトペン・エミュレーションの OFF
000FH	マウスの移動比率の設定
0010H	カーソル消去範囲の設定
0014H	割り込みサブルーチンの交換
0015H	状態退避バッファのサイズの取得
0016H	ドライバ状態の退避
0017H	ドライバ状態の復帰
0018H	代替割り込みサブルーチンの設定
0019H	割り込みサブルーチンのアドレス取得
001AH	マウスの感度の設定
001BH	マウスの感度の取得
001DH	CRT ページ番号の設定
001EH	CRT ページ番号の取得
001FH	マウス BIOS の使用不可
0020H	マウス BIOS の使用可
0021H	ソフトウェア・リセット

以下、マウス BIOS (INT33H) の機能を各番号別に示します。

▶ INT33H(0000H) マウスの初期化		
入力パラメータ AX=0000H		リターン情報 AX=マウスの状態 0 : マウス使用不可能 -1 : マウス使用可能 BX=マウスのボタンの個数
機 能 マウスの使用を開始するには必ずこの機能呼び出す必要があります。 マウスの使用を終了する際にも呼び出す必要があります。 また、この機能により「カーソル消去範囲の指定」が無効になります。 マウスの初期設定は次のとおりです。		
カーソル	状態	非表示
	位置	画面中央
	指示点	(0, 0)
	形状	ビデオ・モードに依存
マウス	X方向の移動率	8 : 8
	Y方向の移動率	16 : 8
割り込みマスク		すべて 0

▶ INT33H(0001H) カーソルの表示		
入力パラメータ AX=0001H		リターン情報 なし
機 能 マウス BIOS では、表示のための内部フラグをもっており、初期値は-1 に設定されています。表示の指示があった場合、マウス BIOS はこのフラグを+1 して(ただし 0 のときは+1 しない)、その結果が 0 の場合のみカーソルの表示を行っています。 これにより、あるプログラムで復帰時に表示の機能が呼ばれても、元のプログラムでは非表示のまま呼ばれていた場合には、グラフィックの破壊などを起こさない、というような制御が可能となっています。		



▶ INT33H(0002H) カーソルの消去	
入力パラメータ AX=0002H	リターン情報 なし
機 能 この機能では、表示のための内部フラグはつねに-1 されます。 画面にグラフィックを表示する場合は、カーソルを破壊するおそれがあるので、必ずこのカーソルの消去を行わなければなりません。	

▶ INT33H(0003H) カーソル位置とボタン情報の取得	
入力パラメータ AX=0003H	リターン情報 BX=ボタンの情報 ビット1：右ボタン ビット0：左ボタン CX=X 座標 DX=Y 座標
機 能 現在のマウスの情報を取得します。	

▶ INT33H(0004H) カーソルの移動	
入力パラメータ AX = 0004H CX = X 座標 DX = Y 座標	リターン情報 なし
機 能 カーソルを指定位置に移動させます。	

▶ INT33H(0005H) ボタンが押された回数と最終位置の取得	
入力パラメータ AX = 0005H BX = ボタンの指定 0 : 左ボタン 1 : 右ボタン	リターン情報 AX = ボタンの情報 ビット 1 : 右ボタン ビット 0 : 左ボタン BX = ボタンが押された回数 CX = 最後に押された X 座標 DX = 最後に押された Y 座標
機 能 この機能が呼び出されると、押された回数は 0 に初期化されます。	

▶ INT33H(0006H) ボタンが離された回数と最終位置の取得	
入力パラメータ AX = 0006H BX = ボタンの指定 0 : 左ボタン 1 : 右ボタン	リターン情報 AX = ボタンの情報 ビット 1 : 右ボタン ビット 0 : 左ボタン BX = ボタンが離された回数 CX = 最後に離された X 座標 DX = 最後に離された Y 座標
機 能 この機能が呼び出されると、離された回数は 0 に初期化されます。	

▶ INT33H(0007H) カーソルの X 方向の移動範囲設定	
入力パラメータ AX=0007H CX = 移動範囲の左端の座標 DX = 移動範囲の右端の座標	リターン情報 なし
機 能 この機能が呼び出された後は、カーソルはこの範囲のみ移動します。 この機能で設定した時点でカーソルが範囲外にある場合は、強制的に範囲境界に移動させます。	

▶ INT33H(0008H) カーソルの Y 方向の移動範囲設定	
入力パラメータ AX=0008H CX = 移動範囲の上端の座標 DX = 移動範囲の下端の座標	リターン情報 なし
機 能 この機能が呼び出された後は、カーソルはこの範囲のみ移動します。 この機能で設定した時点でカーソルが範囲外にある場合は、強制的に範囲境界に移動させます。	

▶ INT33H(0009H) グラフィック・カーソルの形状設定	
入力パラメータ AX=0009H BX = 指示点の X 座標 CX = 指示点の Y 座標 ES: DX = パターン・アドレス	リターン情報 なし
機 能 パターンは AND マスク (32 バイト) と XOR マスク (32 バイト) から構成されます。 指示点は、パターンの左上を原点として -128 から +127 の範囲で指定可能です。	

▶ INT33H(000AH) テキスト・カーソルの形状設定	
入力パラメータ AX=000AH BX = カーソルの実現法	リターン情報 なし

0: ソフトウェア 1: ハードウェア CX = AND マスクの値 DX = XOR マスクの値	
機 能 パターンは AND マスク (32 バイト) と XOR マスク (32 バイト) から構成されます。 指示点は、パターンの左上を原点として -128 から +127 の範囲で指定可能です。	

▶ INT33H(000BH) カーソルの相対移動距離の取得	
入力パラメータ AX = 000BH	リターン情報 CX = X 方向のマウス△ DX = Y 方向のマウス△
機 能 最後にこの機能が呼び出されてからの相対距離を、マウス△単位で返します。	

▶ INT33H(000CH) 割り込みサブルーチンの設定	
入力パラメータ AX = 000CH CX = 割り込み要因マスク ES: DX = 割り込みアドレス	リターン情報 なし
機 能 指定した要因が発生した場合に、所定のサブルーチンへ割り込むように設定します。 割り込み要因マスクには次の値を設定します。 ビット 4: 右ボタンが離された ビット 3: 押された ビット 2: 左ボタンが離された ビット 1: 押された ビット 0: カーソル位置が変更された 割り込みがかかった際には、レジスタには次の値が格納されています。 • AX = マウスの状態 ビット 4: 右ボタンが離された ビット 3: 押された ビット 2: 左ボタンが離された	

ビット 1：左ボタンが押された
ビット 0：カーソル位置が変更された
・ BX = ボタンの状態
ビット 1：右ボタンが押されている
ビット 0：左ボタンが押されている
・ CX = カーソルの X 座標
・ DX = カーソルの Y 座標
・ SI = X 方向のマウス△
・ DI = Y 方向のマウス△

▶ INT33H(000DH) ライトペン・エミュレーションの ON	
入力パラメータ AX = 000DH	リターン情報 なし
機 能 マウスでライトペンのエミュレーションを行います。 カーソルの位置でライトペンの位置を示し、マウスの両方のボタンを押すと、ライトペンを押し下げたことになります。	

▶ INT33H(000EH) ライトペン・エミュレーションの OFF	
入力パラメータ AX = 000EH	リターン情報 なし
機 能 ライトペン・エミュレーションを不可にします。	

▶ INT33H(000FH) マウスの移動比率の設定	
入力パラメータ AX = 000FH CX = X 方向のマウス△ DX = Y 方向のマウス△	リターン情報 なし
機 能 仮想画面座標上で、8 だけ動かすのに必要なマウスの移動距離を設定します。	

▶ INT33H(0010H) カーソル消去範囲の設定	
入力パラメータ AX=0010H CX=左上の X 座標 DX=左上の Y 座標 SI =右下の X 座標 DI =右下の Y 座標	リターン情報 なし
機 能 アプリケーションでグラフィックを描画する際に、描画範囲をこの機能で指定しておくと、それ以外の領域でのカーソルの消去を行いません。単純なカーソルの消去に比べ、処理にかかる時間が短縮されます。 この範囲指定は、「カーソル」の表示により無効となります。	

▶ INT33H(0014H) 割り込みサブルーチンの交換	
入力パラメータ AX=0014H CX=割り込み要因マスク ES:DX=割り込みアドレス	リターン情報 CX=元の割り込み要因マスク ES:DX=元の割り込みアドレス
機 能 すでに設定されている要因の割り込み先を交換して、元の情報を取得します。 割り込み要因マスクには次の値を設定します。 <p> ビット 4: 右ボタンが離された ビット 3: 押された ビット 2: 左ボタンが離された ビット 1: 押された ビット 0: カーソル位置が変更された </p> 割り込みがかかった際には、レジスタには次の値が格納されています。 <p> ・AX=マウスの状態 ビット 4: 右ボタンが離された ビット 3: 押された ビット 2: 左ボタンが離された ビット 1: 押された ビット 0: カーソル位置が変更された </p>	

- BX = ボタンの状態
 ビット 1 : 右ボタンが押されている
 ビット 0 : 左ボタンが押されている
- CX = カーソルの X 座標
- DX = カーソルの Y 座標
- SI = X 方向のマウスΔ
- DI = Y 方向のマウスΔ

▶ INT33H(0015H) 状態退避バッファのサイズの取得

入力パラメータ AX = 0015H	リターン情報 BX = 必要なバッファ・サイズ
機能 ドライバ状態の退避(INT33H, 0016H)で必要な状態退避バッファのサイズを取得します。	

▶ INT33H(0016H) ドライバ状態の退避

入力パラメータ AX = 0016H ES : DX = 状態退避バッファ	リターン情報 なし
機能 現在のマウス・ドライバの状態をバッファに退避します。	

▶ INT33H(0017H) ドライバ状態の復帰

入力パラメータ AX = 0017H ES : DX = 状態退避バッファ	リターン情報 なし
機能 バッファに退避されていたマウス・ドライバの状態を復帰させます。	

▶ INT33H(0018H) 代替割り込みサブルーチンの設定

入力パラメータ AX = 0018H CX = 割り込み要因マスク ES : DX = 割り込みアドレス	リターン情報 なし
---	--------------

機 能

指定した要因が発生した場合に、所定のサブルーチンへ割り込むように設定します。割り込み要因マスクにキー・ストロークが加わったこと以外は、「割り込みサブルーチンの設定」(INT33H, 000CH)と同様の機能です。

割り込み要因マスクには次の値を設定します。

- ビット 7: ボタンの操作とともに Alt キーが押された
- ビット 6: ボタンの操作とともに Ctrl キーが押された
- ビット 5: ボタンの操作とともに Shift キーが押された
- ビット 4: 右ボタンが離された
- ビット 3: 押された
- ビット 2: 左ボタンが離された
- ビット 1: 押された
- ビット 0: カーソル位置が変更された

割り込みがかかった際には、レジスタには次の値が格納されています。

- AX=マウスの状態

- ビット 4: 右ボタンが離された
- ビット 3: 押された
- ビット 2: 左ボタンが離された
- ビット 1: 押された
- ビット 0: カーソル位置が変更された

- BX=ボタンの状態

- ビット 1: 右ボタンが押されている
- ビット 0: 左ボタンが押されている

- CX=カーソルの X 座標

- DX=カーソルの Y 座標

- SI =X 方向のマウスΔ

- DI =Y 方向のマウスΔ

▶ INT33H(0019H) 割り込みサブルーチンのアドレス取得

入力パラメータ

AX=0019H

CX=割り込み要因マスク

リターン情報

AX=設定状況

-1: 設定されていない

CX=元の割り込み要因マスク

BX: DX=元の割り込みアドレス

機 能

指定した要因に対する、代替割り込みサブルーチンのアドレスを取得します。
割り込み要因マスクには次の値を設定します。

ビット 7: ボタンの操作とともに Alt キーが押された
 ビット 6: ボタンの操作とともに Ctrl キーが押された
 ビット 5: ボタンの操作とともに Shift キーが押された
 ビット 4: 右ボタンが離された
 ビット 3: 押された
 ビット 2: 左ボタンが離された
 ビット 1: 押された
 ビット 0: カーソル位置が変更された

▶ INT33H(001AH) マウスの感度の設定**入力パラメータ**

AX=001AH

BX=X 方向の移動比率係数

CX=Y 方向の移動比率係数

リターン情報

なし

機 能

それぞれの移動比率係数は 1 ~ 100 までが設定可能で、初期値は 50 です。

▶ INT33H(001BH) マウスの感度の取得**入力パラメータ**

AX=001BH

リターン情報

BX=X 方向の移動比率係数

CX=Y 方向の移動比率係数

機 能

それぞれの移動比率係数は 1 ~ 100 までで、初期値は 50 です。

▶ INT33H(001DH) CRT ページ番号の設定**入力パラメータ**

AX=001DH

BX=表示 CRT ページ番号

リターン情報

なし

機 能

カーソルを表示する CRT ページを設定します。

ページ番号は 2 色モードの場合と 16 色モードの場合で異なります。

・2色モードの場合

CRT ページ番号は1しかないので、意味をもちません。

・16色モードの場合

CRT ページ番号は1から15まで指定可能で、次の意味をもちます。

ビット3：Iバンク

ビット2：Rバンク

ビット1：Gバンク

ビット0：Bバンク

▶ INT33H(001EH) CRT ページ番号の取得

入力パラメータ

AX=001EH

リターン情報

BX=表示 CRT ページ番号

機能

カーソルを表示する CRT ページを取得します。

ページ番号は2色モードの場合と16色モードの場合で異なります。

・2色モードの場合

CRT ページ番号は1しかないので、意味をもちません。

・16色モードの場合

CRT ページ番号は1から15まで指定可能で、次の意味をもちます。

ビット3：Iバンク

ビット2：Rバンク

ビット1：Gバンク

ビット0：Bバンク

▶ INT33H(001FH) マウス BIOS の使用不可

入力パラメータ

AX=001FH

リターン情報

AX=復元状況

-1：使用不可にできない

ES：BX=元の INT33H

機能

マウス BIOS が設定していたすべての割り込みベクトルを、元の状態に復元します。

ES：BX の値は INT33H を復元するのに使用できます。

▶ INT33H(0020H) マウス BIOS の使用可	
入力パラメータ AX=0020H	リターン情報 なし
機 能 マウス BIOS が使用する割り込みベクタを設定します。	

▶ INT33H(0021H) ソフトウェア・リセット	
入力パラメータ AX=0021H	リターン情報 AX=マウス BIOS 導入状態 -1 : 導入されている 33H : 導入されていない BX=マウスのボタンの数
機 能 この機能は、マウスのハードウェアを初期化していない点を除き、「マウスの初期化」(INT33H, 0000H)と同様です。	

第 6 章

V-Text

V-Text は、IBM-PC の特徴でもある多種のビデオ・ボードのもつ機能を、日本 IBM が正式に規格化したものです。

これにより、より高品位な文字フォントを使った「高品位モード」と、1 画面内により多くの文字を表示する「高密度モード」が実現されました。

V-Text に対応するためには「DOS/V 拡張キット」に含まれる、V-Text 対応ドライバをインストールする必要がありますが、これによりビデオ BIOS に表 6-1 のような機能が拡張されます。

表 6-1 V-Text により拡張・追加されるビデオ BIOS

機能番号	機能内容
00H	ビデオ・モードの設定
1118H	高密度文字フォントの切り替え
1131H	拡張モード・テーブルの取得
12-38H	文字フォント・サイズの切り替え
12-39H	文字フォント密度の切り替え
12-3AH	拡張モード情報の取得

V-Text 対応のアプリケーションを開発する場合は、これらのビデオ BIOS を利用します。

V-Text のディスプレイ・ドライバはドライバ自体が将来登場するであろう各種のアクセラレータに対応できるように、汎用的な「共通サブ・システム」とアクセラレータに依存した「ビデオ拡張ドライバ」に分け、最小限度の拡張で対応が可能になるように設計されています。

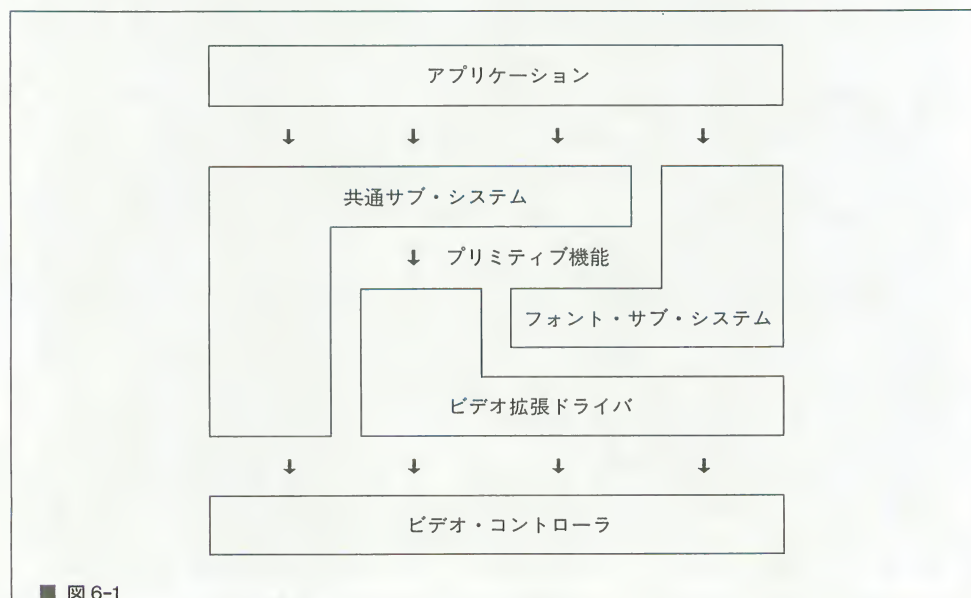


図 6-1 V-Text 対応ドライバの構造

共通サブシステムは、次のようなハードウェア依存しない部分を担当しています。

- ビデオ BIOS の機能の実現
- ビデオ拡張ドライバの登録と解除
- 論理テキスト・バッファの管理
- 論理カーソルの管理
- 論理パレットの管理
- ビデオ BIOS 機能からプリミティブ機能への変換

また、ビデオ拡張ドライバは、ハードウェアに密着した次のような機能を実現しています。これらはプリミティブ機能と呼ばれます。

- 拡張テキスト・モードの設定と解除
- 文字フォント・イメージの表示
- カーソル・イメージの表示と消去
- カラー・パレットの変更
- VRAM のスクロールと塗りつぶし
- フォントの変換
- ビデオ拡張ドライバの状態の退避と回復

6.1 共通サブ・システムとのインタフェイス

V-Text には、ハードウェアに密着したビデオ拡張ドライバを、共通サブ・システムと結びつけて拡張させるためのインタフェイスが用意されています。

表 6-2 拡張インタフェイス(INT15H)機能一覧

機能番号	機能内容
5010H	ビデオ拡張情報の取得
5011H	ビデオ拡張機能の登録
5012H	ビデオ拡張ドライバの解除
5013H	ビデオ拡張ドライバのロック設定

以下、拡張インタフェイス(INT15H)の機能を各番号別に示します。

▶ INT15H(5010H) ビデオ拡張情報の取得		
入力パラメータ AX=5010H		リターン情報 CF=0 の場合 AH=00H の場合 正常終了 ES: BX=情報テーブル・アドレス AH=86H 機能がサポートされていない CF=1 の場合 エラー終了
機 能 情報テーブルの内容		
オフセット	バイト	内 容
00H	1 バイト	共通サブ・システムのメジャー・バージョン番号
01H	1 バイト	共通サブ・システムのマイナー・バージョン番号
02H	1 ワード	拡張ビデオ情報 ビット 1: 拡張テキスト・モード状態 ビット 0: ビデオ拡張ドライバの登録
04H	1 ワード	テキスト・バッファのセグメント
06H	1 ワード	テキスト・バッファのバイト・サイズ
08H	2 ワード	現在のビデオ拡張ドライバの情報テーブルへの FAR ポインタ
0CH	1 ワード	ロック・カウンタ

0EH	2 ワード	元のビデオ BIOS(INT10H)のエントリ・アドレス
12H	1 ワード	フォント・サイズ設定テーブルへの NEAR ポインタ
14H	1 ワード	テキスト密度設定テーブルへの NEAR ポインタ
16H	1 ワード	パレットとオーバースキャンの退避領域への NEAR ポインタ
18H	1 バイト	現行の拡張テキスト・モードのインデックス

・ロック・カウンタ

現在ビデオ拡張ドライバを利用しているプログラムの数を示しています。したがって、この値が0 でなければビデオ拡張ドライバは解除できません。

共通サブ・システムでは、この値を、拡張テキスト・モードに設定を行う際に+1、基本テキスト・モードに戻した場合に-1 しています。

・フォント・サイズ設定テーブルとテキスト密度設定テーブル

これらのテーブルは、次の構造をもちます。モード・インデックスは「ビデオ拡張機能の登録」(INT15H, AX=5011H)で登録される拡張モード・テーブルのインデックスで、初期値は-1 です。

オフセット	バイト数	内 容
00H	1 ワード	エントリ数
01H	2 バイト	ビデオ・モード番号(1 バイト) モード・インデックス(1 バイト)
03H	2 バイト	⋮
⋮	⋮	⋮

・パレットとオーバースキャンの退避領域

この領域は、次の構造をもちます。

オフセット	バイト数	内 容
00H	1 バイト	パレット 0
01H	1 バイト	パレット 1
⋮	⋮	⋮
0FH	1 バイト	パレット 15
10H	1 バイト	オーバースキャン

▶ INT15H(5011H) ビデオ拡張機能の登録

入力パラメータ

AX=5011H

ES: BX=テーブル・アドレス

リターン情報

CF=0 の場合

正常終了

CF=1 の場合

エラー終了

AH=エラー・ステータス

機 能

・ テーブルの内容

オフセット	バイト	内 容
00H	1 バイト	共通サブ・システムのメジャー・バージョン番号
01H	1 バイト	共通サブ・システムのマイナー・バージョン番号
02H	1 ワード	プリミティブ機能テーブルへの NEAR ポインタ
04H	1 ワード	拡張モード・テーブルへの NEAR ポインタ
06H	1 ワード	拡張モード・テーブルのエントリ数
08H	2 ワード	ビデオ拡張ドライバ固有文字列への NEAR ポインタ(文字長制限なし, 00H で終了)

・ バージョン番号

これは、ビデオ拡張ドライバが、どのレベルの DOS/V ビデオ拡張仕様に一致しているかを示しています。共通サブ・システムや他のシステム・プログラムはこの値を参照して、プリミティブ機能の種類を判定するので、現行の DOS/V ビデオ拡張仕様書に示されているバージョン番号と一致させなければなりません。

バージョン番号は、1993 年 3 月現在、次の値となっています。

メジャー・バージョン番号 : 01H

マイナー・バージョン番号 : 01H

・ プリミティブ機能テーブルの内容

オフセット	バイト	NEAR ポインタで示される機能
00H	1 ワード	SetToExtVideoMode
02H	1 ワード	ResetToVGA Mode
04H	1 ワード	WriteSBCSChar
06H	1 ワード	WriteDBCSChar
08H	1 ワード	FillRectangle
0AH	1 ワード	ScrollUp
0CH	1 ワード	ScrollDown
0EH	1 ワード	SetCursor
10H	1 ワード	PutCursor
12H	1 ワード	EraseCursor
14H	1 ワード	SetPalette

16H	1 ワード	ChangeFont
18H	1 ワード	ReturnStateSize
1AH	1 ワード	SaveState
1CH	1 ワード	RestoreState

・拡張モード・テーブルの内容(1 エントリは 16 バイト)

オフセット	バイト	内 容
00H	1 バイト	ビデオ・モード番号
01H	1 バイト	ビデオ・モード情報 ビット 7: モード・エントリ無効 ビット 1: スクロール時に FillRectangle を呼ぶ ビット 0: ハードウェア・カーソル
02H	1 バイト	画面の桁数
03H	1 バイト	画面の行数
04H	1 バイト	文字セルの幅
05H	1 バイト	文字セルの高さ
06H	1 バイト	文字フォントの幅
07H	1 バイト	文字フォントの高さ
08H	1 ワード	水平画面解像度
0AH	1 ワード	垂直画面解像度
0CH	1 ワード	予約済み
0EH	1 ワード	予約済み

・エラー・ステータス

ステータス	内 容
00H	正常終了
01H	すでにビデオ拡張ドライバが登録済み
02H	テキスト・バッファのサイズが拡張モード・テーブルから計算される 値よりも小さい
03H	ビデオ拡張ドライバの登録拒否
86H	機能がサポートされていない

▶ INT15H(5012H) ビデオ拡張ドライバの解除											
入力パラメータ AX=5012H	リターン情報 CF=0 の場合 正常終了 CF=1 の場合 エラー終了 AH=エラー・ステータス										
機 能 ・エラー・ステータス <table border="1"> <thead> <tr> <th>ステータス</th><th>内 容</th></tr> </thead> <tbody> <tr> <td>00H</td><td>正常終了</td></tr> <tr> <td>01H</td><td>ビデオ拡張ドライバが登録されていない</td></tr> <tr> <td>02H</td><td>ビデオ拡張ドライバがロックされている</td></tr> <tr> <td>86H</td><td>機能がサポートされていない</td></tr> </tbody> </table>		ステータス	内 容	00H	正常終了	01H	ビデオ拡張ドライバが登録されていない	02H	ビデオ拡張ドライバがロックされている	86H	機能がサポートされていない
ステータス	内 容										
00H	正常終了										
01H	ビデオ拡張ドライバが登録されていない										
02H	ビデオ拡張ドライバがロックされている										
86H	機能がサポートされていない										

▶ INT15H(5013H) ビデオ拡張ドライバのロック設定							
入力パラメータ AX=5013H BX=ロック状態 1 : ロック -1 : ロック解除	リターン情報 CF=0 の場合 正常終了 CF=1 の場合 エラー終了 AH=エラー・ステータス						
機 能 ・エラー・ステータス <table border="1"> <thead> <tr> <th>ステータス</th><th>内 容</th></tr> </thead> <tbody> <tr> <td>00H</td><td>正常終了</td></tr> <tr> <td>86H</td><td>機能がサポートされていない</td></tr> </tbody> </table>		ステータス	内 容	00H	正常終了	86H	機能がサポートされていない
ステータス	内 容						
00H	正常終了						
86H	機能がサポートされていない						

6.2 プリミティブ機能

プリミティブ機能は、拡張ビデオ・ドライバのハードウェアに依存する部分の機能のみを集めたものです。この機能を実現できるならば、ディスプレイ・ドライバはいかなるハードウェア上でも動作可能となります。

表 6-3 プリミティブ機能一覧

番号	機能名	機能内容
0	SetToExtVideoMode	拡張ビデオ・モードの設定
1	ResetToVGAMode	VGA モードへの復帰
2	WriteSBCSChar	半角文字の描画
3	WriteDBCSChar	全角文字の描画
4	FillRectangle	矩形の描画
5	ScrollUp	上方向へのスクロール
6	ScrollDown	下方向へのスクロール
7	SetCursorShape	カーソル形状の設定
8	PutCursor	カーソルの表示
9	EraseCursor	カーソルの消去
10	SetPalette	パレットの設定
11	ChangeFont	半角文字フォントの変更
12	ReturnStateSize	ステートサイズの取得
13	SaveStateSize	ステートサイズの退避
14	RestoreStateSize	ステートサイズの復帰

以下、プリミティブ機能を各番号別に示します。

機能 0 : SetToExtVideoMode 拡張ビデオ・モードの設定	
入力パラメータ AL = 拡張モード・テーブルのインデックス AH = 00H	リターン情報 なし
機 能 ビデオ BIOS の、ビデオ・モードの設定 (INT10H, AH=00H) から呼び出されます。 この機能が呼び出されてから、次に機能 1 の「ResetToVGAMode」が呼び出されるまでは、独自の拡張ビデオ・モードになります。 ここでは次の状態を設定します。 <ul style="list-style-type: none"> ・必要な内部変数の値 ・パレット情報 (現行値の取得) ・カーソル非表示 	

▶機能 1: ResetToVGA Mode VGA モードへの復帰	
入力パラメータ	リターン情報
なし	なし
機 能 ビデオ BIOS の、ビデオ・モードの設定 (INT10H, AH=00H) から呼び出されます。 標準の VGA モードへ復帰します。この機能から復帰後に共通サブ・システムが VGA のレジスタ設定を行うので、ここで行うのは最小限度の操作だけで構いません。	

▶機能 2: WriteSBCSChar 半角文字の描画	
入力パラメータ	リターン情報
AL = 属性 1 AH = 属性 2 CH = 00H CL = 半角文字コード DH = 行位置 DL = 桁位置	なし
機 能 ビデオ BIOS の、文字列表示関連の機能から呼び出されます。 半角文字を描画します。 3 バイト属性モードでなければ、属性 2 は無効です。	

▶機能 3: WriteDBCSChar 全角文字の描画	
入力パラメータ	リターン情報
AL = 第 1 バイトの属性 1 AH = 第 1 バイトの属性 2 BL = 第 2 バイトの属性 1 BH = 第 2 バイトの属性 2 CX = 全角文字コード DH = 行位置 DL = 桁位置	なし
機 能 ビデオ BIOS の、文字列表示関連の機能から呼び出されます。 全角文字を描画します。 3 バイト属性モードでなければ、属性 2 は無効です。 この機能を使って、画面の最右端に描画が行われることはありません。この場合は、テキスト・バッファには指定どおりの書き込みが行われますが、画面表示は共通サブ・システムにより機能 2 の「WriteSBCSChar」が呼び出され、半角スペースが描画されます。	

機能 4 : FillRectangle 矩形の描画

入力パラメータ	リターン情報
AL = 塗りつぶす色 上位 4 ビット : 文字色 下位 4 ビット : 背景色 CH = 左上の行位置 CL = 桁位置 DH = 右上の行位置 DL = 桁位置	なし
機 能 ビデオ BIOS の、上下スクロール (INT10H, AH=06H/07H) や、プリミティブ機能の 6/7 (ScrollUp, ScrollDown) から呼び出されます。 この機能は、全画面のスクロール (消去) を行ったり、部分的スクロールの後のブランク行の消去に使用されます。 通常は背景色で塗りつぶしますが、ハードウェア・テキスト・バッファをもつ場合は、指定の文字色を設定しておきます。	

機能 5 : ScrollUp 上方向へのスクロール

入力パラメータ	リターン情報
AL = スクロールする行数 (0 はありえない) AH = 塗りつぶす色 上位 4 ビット : 文字色 下位 4 ビット : 背景色 CH = 左上の行位置 CL = 桁位置 DH = 右上の行位置 DL = 桁位置	なし
機 能 ビデオ BIOS の、上スクロール (INT10H, AH=06H) から呼び出されます。 モード情報のビット 1 が 0 の場合は、指定された領域を上スクロールさせるだけで、スクロール後のブランク行を消去する必要はありません。ブランク行の消去は機能 4 の「FillRectangle」が行います。 ビット 1 が 1 の場合は、指定された領域を上スクロールさせると同時に、ブランク行を塗りつぶす必要があります。 スクロールの際にハードウェア・スクロール機能を使用するかどうかは、この機能内部で判断します。	

機能 6 : ScrollDown 下方向へのスクロール	
入力パラメータ AL = スクロールする行数 (0 はありえない) AH = 塗りつぶす色 上位 4 ビット : 文字色 下位 4 ビット : 背景色 CH = 左上の行位置 CL = 桁位置 DH = 右上の行位置 DL = 桁位置	リターン情報 なし
機 能 ビデオ BIOS の、下スクロール (INT10H, AH=07H) から呼び出されます。 モード情報のビット 1 が 0 の場合は、指定された領域を下にスクロールさせるだけで、スクロール後のブランク行を消去する必要はありません。ブランク行の消去は機能 4 の「FillRectangle」が行います。 ビット 1 が 1 の場合は、指定された領域を下にスクロールさせると同時にブランク行を塗りつぶす必要があります。 スクロールの際にハードウェア・スクロール機能を使用するかどうかは、この機能内部で判断します。	

機能 7 : SetCursorShape カーソル形状の設定	
入力パラメータ CH = 開始位置 CL = 終了位置	リターン情報 なし
機 能 ビデオ BIOS の、「カーソル形状の設定」(INT10H, AH=01H) から呼び出されます。 カーソルの高さの上限値は、CGA の 8 ではなく、「拡張モード・テーブル」に指定されている文字フォントの高さです。この比率の変換は共通サブ・システムが行います。 この機能では表示・非表示の状態を変えてはいけません。	

機能 8 : PutCursor カーソルの表示	
入力パラメータ AL = 表示色 DH = 行位置 DL = 桁位置	リターン情報 なし
機 能 カーソルの表示が必要な場合に、随時呼び出されます。 カーソルの色は文字色と同じであることが必要ですが、正確なエミュレーションが困難な場合は、イメージの反転などの処理でも可能です。	

機能 9 : EraseCursor カーソルの消去	
入力パラメータ なし	リターン情報 なし
機 能 カーソルの消去が必要な場合に、随時呼び出されます。 カーソルの表示を消去します。	

機能 10 : SetPalette パレットの設定	
入力パラメータ AL = パレット番号 AH = カラー・レジスタ番号	リターン情報 なし
機 能 ビデオ BIOS の、パレット設定 (INT10H, AH=10H, AL=00H/01H/02H) から呼び出されます。 オーバースキャンはパレット番号 16 を指定します。この機能では、カラー・レジスタ番号を対応するハードウェアの最も適切な値に変換して設定する必要があります。	

▶ 機能 11 : ChangeFont 半角文字フォントの変更

入力パラメータ AL = 半角文字コード BH = 文字フォントの幅 BL = 文字フォントの高さ ES : SI = バッファ・アドレス	リターン情報 なし
機 能 ビデオ BIOS の、「ユーザ定義の文字フォント登録」(INT10H, AX=1100H) から呼び出されます。 指定されたフォント・パターンを文字生成バッファ内の対応する領域にロードしますが、この際に必要ならばサイズ変更を行ってもかまいません。フォント・パターンはビット列で、隙間なく連続して格納されています。 フォント・パターンのロードだけを行い、画面への再表示を行ってはいけません。これは共通サブ・システムが指示を行います。	

▶ 機能 12 : ReturnStateSize ステートサイズの取得

入力パラメータ AL = 要求ステート ビット 1 : 半角文字生成バッファ ビット 0 : 内部変数	リターン情報 CX = バッファ・サイズ
機 能 共通サブ・システムから呼び出されます。 要求のあった内部変数および半角文字生成バッファのサイズを返します。	

▶ 機能 13 : SaveStateSize ステートサイズの退避

入力パラメータ AL = 要求ステート ビット 1 : 半角文字生成バッファ ビット 0 : 内部変数 ES : BX = 退避先バッファ	リターン情報 なし
機 能 共通サブ・システムから呼び出されます。 要求のあった内部変数および半角文字生成バッファの内容を退避します。	

▶ 機能 14 : RestoreStateSize ステートサイズの復帰	
入力パラメータ AL = 要求ステート ビット 1 : 半角文字生成バッファ ビット 0 : 内部変数 ES : BX = 復帰元バッファ	リターン情報 なし
機 能 共通サブ・システムから呼び出されます。 要求のあった内部変数および半角文字生成バッファの内容を復帰します。	

6.3 ビデオ拡張プロファイル

ビデオ拡張ドライバを作成した場合は、インストール・プログラムのために、それに対応したビデオ拡張プロファイルを作成しなければなりません。

このビデオ拡張プロファイルは、ビデオ拡張ドライバと同一ファイル名で拡張子名が「.PRO」でなければなりません(インストールされる場合は、これは DSPX.PRO というファイル名でコピーされます)。

ビデオ拡張プロファイルは、次のような構造の複数のエントリから構成されるバイナリ・データです。終端はエントリ長に 0(1 ワード)を設定します。

エントリ長 : 1 ワード(エントリ長を含む)
エントリ ID : 16 バイト固定
エントリ内容 : 可変長

記述されるエントリは次のとおりです。

表 6-4 ビデオ拡張プロファイルのエントリ一覧

エントリ長	48(1 ワード)
エントリ ID	"VideoCardInfo"
エントリ内容	サポートするビデオの名前(30 バイト)
エントリ長	30(1 ワード)
エントリ ID	"DriverFileName"
エントリ内容	拡張ドライバのファイル名(12 バイト)
エントリ長	2+16+2+2+モード・テーブル長(1 ワード)
エントリ ID	"VideoModeTable"
エントリ内容	ビデオ拡張仕様のバージョン(1 ワード) モード・テーブル・エントリ数(1 ワード) モード・テーブル 0(17 バイト) モード・テーブル 1(17 バイト) ⋮ モード・テーブルの構成

オフセット	サイズ	内 容
00H	1 バイト	省略時のモード選択
01H	1 バイト	ビデオ・モード番号
02H	1 バイト	モード情報
03H	1 バイト	桁数
04H	1 バイト	行数
05H	1 バイト	文字セルの幅

	<table><tr><td>06H</td><td>1 バイト</td><td>文字セルの高さ</td></tr><tr><td>07H</td><td>1 バイト</td><td>文字フォントの幅</td></tr><tr><td>08H</td><td>1 バイト</td><td>文字フォントの高さ</td></tr><tr><td>09H</td><td>1 ワード</td><td>画面の水平解像度</td></tr><tr><td>0BH</td><td>1 ワード</td><td>画面の垂直解像度</td></tr><tr><td>0DH</td><td>2 ワード</td><td>予約済み</td></tr></table> <p>最初のバイトが 1 の場合は、現行のモードに従って、「文字フォント・サイズの変更」(INT10H, 12-38H) または「表示文字密度の変更」(INT10H, 12-39H) のいずれかを実行し、システムの省略値のモードを設定します。</p>	06H	1 バイト	文字セルの高さ	07H	1 バイト	文字フォントの幅	08H	1 バイト	文字フォントの高さ	09H	1 ワード	画面の水平解像度	0BH	1 ワード	画面の垂直解像度	0DH	2 ワード	予約済み
06H	1 バイト	文字セルの高さ																	
07H	1 バイト	文字フォントの幅																	
08H	1 バイト	文字フォントの高さ																	
09H	1 ワード	画面の水平解像度																	
0BH	1 ワード	画面の垂直解像度																	
0DH	2 ワード	予約済み																	
エントリ長 エントリ ID エントリ内容	20(1 ワード) "DSPXInfo" DSPX 情報(1 ワード) ・ビット 2 ～ 1 : US モード時のみ 00＝縦長モードが 80×43 01＝縦長モードが 80×50 ・ビット 0 : 0＝簡易表示 1＝詳細表示																		
エントリ長 エントリ ID エントリ内容	22(1 ワード) "DBCSVideoMode" ビデオ・モード番号(1 バイト) 予約済み(1 バイト) 画面の桁数(1 バイト) 画面の行数(1 バイト)																		
エントリ長 エントリ ID エントリ内容	22(1 ワード) "SBCSVideoMode" ビデオ・モード番号(1 バイト) 予約済み(1 バイト) 画面の桁数(1 バイト) 画面の行数(1 バイト)																		
エントリ長 エントリ ID エントリ内容	2+16+2+オプション文字列長(1 ワード) "OptionTable" オプションのエントリ数(1 ワード) オプション文字列 0(64+256 バイト以内) オプション文字列 1(64+256 バイト以内) ⋮ <p>オプション文字列は、オプション文法(64 バイト以内で 0 で終了する文字列)とオプション解説(256 バイト以内で 0 で終了する文字列)から構成されます。 オプション文法は、次の 2 つのタイプのいずれかでなければなりません。</p> <ul style="list-style-type: none">・タイプ A オプション名＝オプション 1/オプション 2…・タイプ B オプション名＝[数字]																		
エントリ長	0000H(1 ワード)																		

List 6-1

拡張ビデオプロファイルの作成例

```

;      Length  "123456789012345678901234567890"

VCI    DW      48
        DB      "VideoCardInfo"
        DB      "SuperVGA(800*600)"

DFN    DW      30
        DB      "DriverFileName"
        DB      "DSPXSVGA.EXE"

VMT    DW      DSPXI-VMT
        DB      "VideoModeTable"
        DW      3
        DB      1, 03H, 00H, 80, 33, 8, 18, 8, 16
        DW      800, 600, 0, 0
        DB      1, 73H, 00H, 80, 33, 8, 18, 8, 16
        DW      800, 600, 0, 0
        DB      0, 70H, 00H, 80, 50, 8, 12, 8, 12
        DW      800, 600, 0, 0

DSPXI  DW      20
        DB      "DSPXInfo"
        DW      0

DBCSVM DW      22
        DB      "DBCSVideoMode"
        DB      03H, 0, 80, 25

SBCSVM DW      22
        DB      "SBCSVideoMode"
        DB      03H, 0, 80, 25

OT     DW      EOP-OT
        DB      "OptionTable"
        DB      "HS=ON/OFF/LC", 0
        DB      "スクロール法を指定します"
        DB      "ON : ハードウェア・スクロール, "
        DB      "OFF: ソフトウェア・スクロール, "
        DB      "LC : LCRを用いたスクロール", 0
        DB      "MODE=[2]", 0
        DB      "スーパーVGAの800×600(16色)を"
        DB      "表示するビデオ・モードを指定します", 0

EOP    DW      0

```

第 7 章

メモリ・システム

DOS/V のメモリ・システムはたいへん複雑です。これは、これまで DOS 自身が全体的な統合を考慮せず、急場しのぎ的にメモリを拡張してきたことが原因です。

さらに最近では、DOS 自身がハイ・メモリ・エリアへの退避を行えるようになりました。DOS の利用環境から考えるとフリー・エリアが広がってたいへんありがたいのですが、こと開発環境からみると、そうでなくとも複雑な事態をいっそう複雑にしまいました。

この状態も、EMS を含めた XMS という規格で比較的統合されてはきていますが、今後本格的なプロテクトモード用の拡張メモリの規格が確定するまでは、当分混沌としていることでしょう。

表 7-1 DOS/V のメモリ・マップ

アドレス	通常の配置	HMA 使用時
0000 : 0000	割り込みベクトル	
0040 : 0000	BIOS ワークエリア	
0050 : 0000	DOS/V ワークエリア	
???? : 0000	IO.SYS または IBMBIO.COM	DOS/V 常駐部 DOS/V バッファ デバイス・ドライバ
	MSDOS.SYS または IBMDOS.COM	COMMAND.COM ユーザ・エリア
	DOS/V バッファ デバイス・ドライバ	
	COMMAND.COM ユーザ・エリア	
FFFF : 0010	未使用	VDISK ヘッド
FFFF : 0030		IO.SYS または IBMBIO.COM
???? : ????		MSDOS.SYS または IBMDOS.COM

7.1 EMS(Expanded Memory Specification)

EMS は 1985 年に、ロータス、インテル、マイクロソフトの 3 社が共同で規格を定めたことから、その頭文字を取って LIM-EMS とも呼ばれ、現在最もメジャーな拡張メモリ方式であり、バージョン 4.0 規格が主流となっています。

EMS は、DOS の 640K バイトのメイン・メモリ外の A000:0000H から FFFF:FFFFH までの間のいずれかの 64K バイトの領域を窓(ページ・フレーム)として使い、この中をさらに 16K バイト単位のバンク(物理ページ)に分けて、ここに実際のメモリ(論理ページ)を割り付けて利用します。これにより都合 8M バイトまでの拡張メモリが利用可能となります。

ページ・フレームがどこの領域に設定されるかは環境によって異なります。

表 7-2 EMS の機能一覧

機能番号	機能内容
40H	ステータスの取得
41H	ページ・フレームの取得
42H	論理ページ数の取得
43H	論理ページの割り付け
44H	論理ページから物理ページへのマッピング
45H	ハンドルの開放
46H	EMM ドライバ・バージョンの取得
47H	ページ・マップの保存
48H	ページ・マップの復元
4BH	ハンドルのカウント数の取得
4CH	ハンドルの論理ページ数の取得
4DH	すべてのハンドルの論理ページ数の取得
4E00H	ページ・マップの取り出し
4E01H	ページ・マップの設定
4E02H	ページ・マップの取り出しと設定
4E03H	ページ・マップの保存領域のサイズ取得
4F00H	指定領域のページ・マップの取り出し
4F01H	指定領域のページ・マップの設定
4F02H	指定領域のページ・マップのサイズ取得
50H	複数の論理ページのマッピング
51H	論理ページの再割り当て
5200H	ハンドル属性の取得 (DOS/V 非サポート)
5201H	ハンドル属性の設定 (DOS/V 非サポート)
5202H	EMM ドライバの属性サポートの取得 (DOS/V 非サポート)
5300H	ハンドル名の取得
5301H	ハンドル名の設定
5400H	すべてのハンドルとハンドル名の取得

5401H	名前つきハンドルの検索
5402H	総ハンドル数の取得
55H	ページ・マップの変更と FAR ジャンプ
56H	ページ・マップの変更と FAR コール
5602H	ページ・マップのスタック・サイズ取得
5700H	メモリ領域の転送
5701H	メモリ領域の交換
5800H	マップ可能なセグメントの取得
5801H	マップ可能なセグメントの項目数の取得
5900H	拡張メモリのハードウェア情報の取得
5901H	Raw ページ数の取得
5A00H	標準サイズのページの割り付け
5A01H	Raw ページの割り付け
5B00H	代替マップ・レジスタ・セットの取得
5B01H	代替マップ・レジスタ・セットの設定
5B02H	代替マップ・レジスタ保存領域のサイズ取得
5B03H	代替マップ・レジスタ・セットの割り付け
5B04H	代替マップ・レジスタ・セットの開放
5B05H	DMA マップ・レジスタ・セットの割り付け
5B06H	DMA を有効にする
5B07H	DMA を無効にする
5B08H	DMA マップ・レジスタ・セットの開放
5CH	ウォーム・ブートの準備
5D00H	OS/E 機能を有効にする
5D01H	OS/E 機能を無効にする
5D02H	アクセス・キーの返還

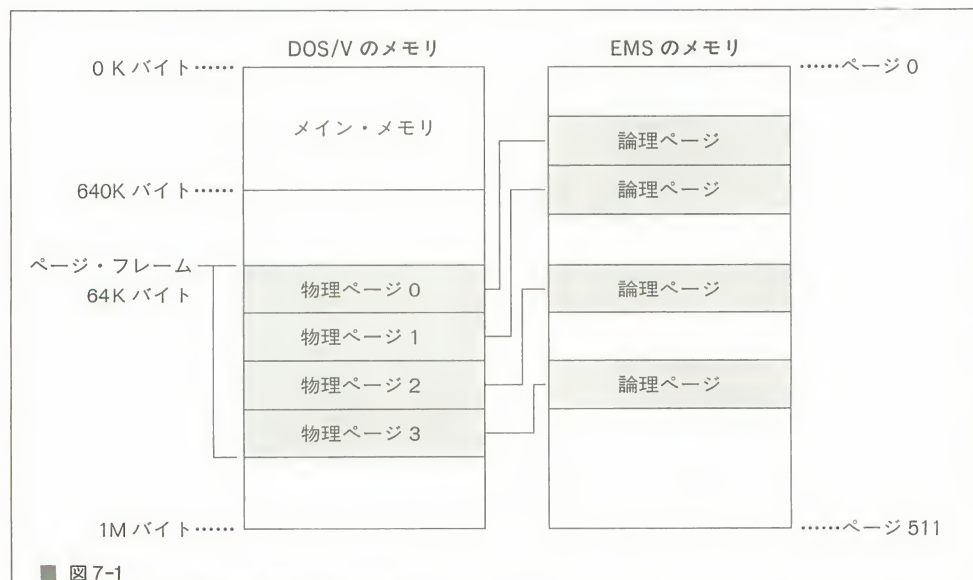


図 7-1

EMS の物理ページと論理ページの関係

EMS の実現方法には、ハード的にバンクを割り付ける物理 EMS 方式や、プロテクト・メモリを利用する仮想 EMS 方式など、いろいろな実現方法があるので、ユーザ側は EMS ドライバが用意したファンクションで操作を行わなければなりません。これらのファンクションは INT67H を呼び出します。

EMS は利用に先だって、次の方法で EMS ドライバ自身が設定されているかどうかを判別しなければなりません。

(1) ファイルのオープン機能を使った方法

ファイル名 " EMMXXXX0 " をオープンしてみます。オープン可能ならば EMS ドライバが常駐しているので、いったんクローズした後、EMS の利用を開始します。この方法は、アプリケーションにとって最も一般的な確認方法です。

List 7-1

EMS の存在確認 1

```

;
; EMSドライバの常駐検査1(キャリーが立てば常駐していない)
;
EXIST_EMS1:
        MOV     AH, 3DH
        MOV     AL, 0
        LEA     DX, EMS_NAME
        INT     21H
        JB      EXIST_EMS1_E
        MOV     BX, AX
        MOV     AH, 3EH
        INT     21H

EXIST_EMS1_E:
        RET

EMS_NAME      DB      "EMMXXXX0", 0

```

(2) 割り込みベクトルを調べる方法

EMS ドライバは INT67H にフックされているので、この割り込みベクトルを取り出します(0000:019CH からの 2 ワード)。

このベクトルからのオフセット 000AH に、ドライバ名 " EMMXXXX0 " が存在しているかどうか調べます。この方法は DOS のファンクションが利用できない可能性のある常駐プログラムやデバイス・ドライバに有効です。

List 7-2

EMS の存在確認 2

```

;
; EMSドライバの常駐検査2(キャリーが立てば常駐していない)
;
EXIST_EMS2:
                XOR     AX,AX
                MOV     EA,AX
                LES     DI,ES:[019CH]
                ADD     DI,000AH
                CLD
                LEA     SI,EMS_NAME

EXIST_EMS2_L:
                LODSB
                OR       AL,AL
                JZ       EXIST_EMS2_E
                INC     DI
                CMP     AL,ES:BYTE PTR [DI-1]
                JZ       EXIST_EMS2_L
                STC

EXIST_EMS2_E:
                RET

EMS_NAME       DB      "EMMXXXX0",0

```

以下、EMS の機能を各番号別に示します。

● EMS のファンクション(バージョン 3.2 互換)

▶ INT67H(40H) ステータスの取得	
入力パラメータ AH=40H	リターン情報 AH=エラー・ステータス
機能 EMM ドライバが使用可能かどうか調べます。	

▶ INT67H(41H) ページ・フレームの取得	
入力パラメータ AH=41H	リターン情報 AH=エラー・ステータス BX=ページフレーム・セグメント
機 能 ページ・フレームを取得します。	

▶ INT67H(42H) 論理ページ数の取得	
入力パラメータ AH=42H	リターン情報 AH=エラー・ステータス BX=未使用の論理ページ数 DX=総論理ページ数
機 能 論理ページの総数や、未使用の論理ページの数を取得します。	

▶ INT67H(43H) 論理ページの割り付け	
入力パラメータ AH=43H BX=割り付ける論理ページ数	リターン情報 AH=エラー・ステータス DX=ハンドル
機 能 標準の 16K バイトの論理ページを割り付けます。返されるハンドルは 1 ～ 254 までです。	

▶ INT67H(44H) 論理ページから物理ページへのマッピング	
入力パラメータ AH=44H AL=物理ページ番号 BX=論理ページ番号 DX=ハンドル	リターン情報 AH=エラー・ステータス
機 能 ハンドルに割り付けられている論理ページのうち、1 ページを物理ページにマッピングします。 論理ページ番号に FFFFH を指定すると、その物理ページに関するマッピングが解除され、論理ページへの読み書きができなくなります。	

▶ INT67H(45H) ハンドルの開放	
入力パラメータ AH=45H DX=ハンドル	リターン情報 AH=エラー・ステータス
機 能 ハンドルの開放を行います。	

▶ INT67H(46H) EMM ドライバ・バージョンの取得	
入力パラメータ AH=46H	リターン情報 AH=エラー・ステータス AL=バージョン(40H)
機 能 バージョンはBCD形式なので、現行のバージョン4.0では40Hとなります(バージョン3.2では32H)。	

▶ INT67H(47H) ページ・マップの保存	
入力パラメータ AH=47H DX=ハンドル	リターン情報 AH=エラー・ステータス
機 能 <p>ページ・マップ・レジスタの内容を、指定したハンドルの内部保存領域に保存します。この機能は、通常ハンドルを使用中の割り込み処理の中で、現在のマッピング状態を保存するために使います。</p> <p>ハンドルは、アプリケーションで使用しているものではなく、割り込み処理が割り当てたものを使用します。</p> <p>この機能では、バージョン3の64Kバイトのページ・フレームのマッピング状態だけを保存します。それ以降のバージョンのページ・マップの保存を行うには、「ページ・マップの取り出し」(INT67H, 4E00H)を使用してください。</p>	

▶ INT67H(48H) ページ・マップの復元	
入力パラメータ AH=48H DX=ハンドル	リターン情報 AH=エラー・ステータス
機 能 ページ・マップの復元を行います。	

▶ INT67H(4BH) ハンドルのカウント数の取得	
入力パラメータ AH=4BH	リターン情報 AH=エラー・ステータス BX=ハンドル数
機 能 使用中のハンドル数には OS 用のハンドル (0) も含みます。	

▶ INT67H(4CH) ハンドルの論理ページ数の取得	
入力パラメータ AH=4CH DX=ハンドル	リターン情報 AH=エラー・ステータス BX=論理ページ数
機 能 ハンドルの論理ページ数を取得します。	

▶ INT67H(4DH) すべてのハンドルの論理ページ数の取得	
入力パラメータ AH=4DH ES:DI=データ構造体	リターン情報 AH=エラー・ステータス BX=使用中のハンドル数
機 能 すべてのハンドルに割り付けられている論理ページ数をデータ構造体に返します。データ構造体の構成は次のようになります。 <div style="margin-left: 40px;"> 1 ワード : ハンドル 1 ワード : 論理ページ数 ⋮ ⋮ </div>	

▶ INT67H(4E00H) ページ・マップの取り出し	
入力パラメータ AX=4E00H ES:DI=ページ・マップ	リターン情報 AH=エラー・ステータス
機 能 格納する領域の大きさは、「ページ・マップの保存領域のサイズ取得」(INT67H, 4E03H) で取得します。	

▶ INT67H(4E01H) ページ・マップの設定	
入力パラメータ AX=4E01H ES:DI=ページ・マップ	リターン情報 AH=エラー・ステータス
機 能 ページ・マップの設定を行います。	

▶ INT67H(4E02H) ページ・マップの取り出しと設定	
入力パラメータ AX=4E02H DS:SI=新ページ・マップ ES:DI=旧ページ・マップ	リターン情報 AH=エラー・ステータス
機 能 現在のページ・マップを旧ページ・マップに取得した後、新ページ・マップを設定します。	

▶ INT67H(4E03H) ページ・マップの保存領域のサイズ取得	
入力パラメータ AX=4E03H	リターン情報 AH=エラー・ステータス AL=ページ・マップのサイズ
機 能 ページ・マップが保存される領域のサイズを取得します。	

● EMS のファンクション(バージョン 4.0 互換)

▶ INT67H(4F00H) 指定領域のページ・マップの取り出し	
入力パラメータ AX=4F00H DS:SI=指定領域 ES:DI=ページ・マップ	リターン情報 AH=エラー・ステータス
機 能 領域指定は次のように行います。 1 ワード : マップ可能なセグメント数 1 ワード : マップ可能なセグメント・アドレス : :	

▶ INT67H(4F01H) 指定領域のページ・マップの設定	
入力パラメータ AX=4F01H DS:DX=ページ・マップ	リターン情報 AH=エラー・ステータス
機 能 指定領域のページ・マップの設定を行います。	

▶ INT67H(4F02H) 指定領域のページ・マップのサイズ取得	
入力パラメータ AX=4F02H BX=指定領域のページ数	リターン情報 AH=エラー・ステータス AL=ページ・マップのサイズ
機 能 指定領域のページ・マップのサイズの取得を行います。	

▶ INT67H(50H) 複数の論理ページのマッピング	
入力パラメータ AH=50H AL=マッピング単位 0:物理ページ 1:セグメント DX=ハンドル CX=ページ数 DS:SI=ページ対応表	リターン情報 AH=エラー・ステータス
機 能 ページ対応表は次の構成を取ります。 <div style="margin-left: 40px;"> 1ワード : 論理ページ番号 1ワード : 物理ページ番号またはセグメント ⋮ ⋮ </div> 論理ページ番号に FFFFH を指定すると、その物理ページのマッピングを解除します。	

▶ INT67H(51H) 論理ページの再割り当て	
入力パラメータ AH=51H DX = ハンドル BX = 再割り当て論理ページ数	リターン情報 AH=エラー・ステータス BX = 割り当て後の論理ページ数
機 能 この機能により、ハンドルに割り当てるページ数を増減できますが、その後の論理ページの内容は保証されません。	

▶ INT67H(5200H) ハンドル属性の取得 (DOS/V 非サポート)	
入力パラメータ AX=5200H DX = ハンドル	リターン情報 AH=エラー・ステータス AL = ハンドル属性 0: 揮発性 1: 不揮発性
機 能 ハンドルが、ウォーム・ブートによる保存が可能かどうかの属性を取得します。	

▶ INT67H(5201H) ハンドル属性の設定 (DOS/V 非サポート)	
入力パラメータ AX=5201H BL = ハンドル属性 0: 揮発性 1: 不揮発性	リターン情報 AH=エラー・ステータス
機 能 ハンドル属性の設定を行います。	

▶ INT67H(5202H) EMM ドライバの属性サポートの取得 (DOS/V 非サポート)	
入力パラメータ AX=5202H	リターン情報 AH=エラー・ステータス AL = 属性のサポート 0: 揮発性のサポートなし 1: 揮発性のサポートあり
機 能 EMM ドライバの属性サポートを取得します。	

▶ INT67H(5300H) ハンドル名の取得	
入力パラメータ AX=5300H DX=ハンドル ES:DI=ハンドル名	リターン情報 AH=エラー・ステータス
機 能 ハンドル名を取得します。	

▶ INT67H(5301H) ハンドル名の設定	
入力パラメータ AX=5301H DX=ハンドル DS:SI=ハンドル名	リターン情報 AH=エラー・ステータス
機 能 ハンドル名は 8 文字で、使用する文字に制限はありませんが、少なくとも 1 文字は NULL 以外の文字を設定しなければなりません。 ハンドル名はいつでも変更可能です。	

▶ INT67H(5400H) すべてのハンドルとハンドル名の取得	
入力パラメータ AX=5400H ES:DI=データ構造体	リターン情報 AH=エラー・ステータス
機 能 データ構造体には 1 ハンドル当たり 10 バイトで、次の構成で格納されています。 1 ワード : ハンドル番号 8 バイト : ハンドル名 : : データ構造体は 2550 バイト(最大値)確保しておく必要があります。	

▶ INT67H(5401H) 名前つきハンドルの検索	
入力パラメータ AX=5401H DS:SI=ハンドル名	リターン情報 AH=エラー・ステータス DX=ハンドル
機 能 名前つきハンドルの検索を行います。	

▶ INT67H(5402H) 総ハンドル数の取得	
入力パラメータ AX=5402H	リターン情報 AH=エラー・ステータス BX=総ハンドル数
機 能 ハンドルの総数を取得します。	

▶ INT67H(55H) ページ・マップの変更と FAR ジャンプ	
入力パラメータ AH=55H AL=マッピング単位 0:物理ページ 1:セグメント DX=ハンドル DS:SI=データ構造体	リターン情報 AH=エラー・ステータス
機 能 ページ・マップを変更した後、指定したアドレスへジャンプします。 データ構造体は次の構成を取ります。 2ワード : ジャンプ先アドレス 1バイト : ページ対応表の長さ(0指定可能) 2ワード : ページ対応表のアドレス	

▶ INT67H(56H) ページ・マップの変更と FAR コール	
入力パラメータ AH=56H AL=マッピング単位 0:物理ページ 1:セグメント DX=ハンドル DS:SI=データ構造体	リターン情報 AH=エラー・ステータス
機 能 ページ・マップを変更した後、指定したアドレスをコールします。コールの前後でマッピングは保存されていますが、そのぶん余分にスタックを消費します。 データ構造体は次の構成を取ります。	

2 ワード	: コール先アドレス
1 バイト	: コール後のページ対応表の長さ (0 指定可能)
2 ワード	: コール後のページ対応表のアドレス
1 バイト	: コール前のページ対応表の長さ (0 指定可能)
2 ワード	: コール前のページ対応表のアドレス
2 ワード	: 予約済み

▶ INT67H(5602H) ページ・マップのスタック・サイズ取得

入力パラメータ

AX=5602H

リターン情報

AH=エラー・ステータス

BX=必要なスタック・サイズ

機 能

このサイズをスタックに加算することで, FAR コール先からマッピングを元に戻さずに復帰することができます。

▶ INT67H(5700H) メモリ領域の転送

入力パラメータ

AX=5700H

DS:SI=データ構造体

リターン情報

AH=エラー・ステータス

機 能

データ構造体の構成は次のようになります。

2 ワード	: 領域のサイズ
1 バイト	: 転送元のメモリ・タイプ
1 ワード	: ハンドル
1 ワード	: オフセット
1 ワード	: ページ
1 バイト	: 転送先のメモリ・タイプ
1 ワード	: ハンドル
1 ワード	: オフセット
1 ワード	: ページ

▶ INT67H(5701H) メモリ領域の交換	
入力パラメータ AX=5701H DS:SI=データ構造体	リターン情報 AH=エラー・ステータス
機 能 データ構造体の構成は次のようになります。 2 ワード : 領域のサイズ 1 バイト : 領域1のメモリ・タイプ 1 ワード : ハンドル 1 ワード : オフセット 1 ワード : ページ 1 バイト : 領域2のメモリ・タイプ 1 ワード : ハンドル 1 ワード : オフセット 1 ワード : ページ	

▶ INT67H(5800H) マップ可能なセグメントの取得	
入力パラメータ AX=5800H DS:SI=データ構造体	リターン情報 AH=エラー・ステータス CX=データの項目数
機 能 データ構造体の構成は次のようになります。 1 ワード : セグメント 1 ワード : 物理ページ ⋮ ⋮	

▶ INT67H(5801H) マップ可能なセグメントの項目数の取得	
入力パラメータ AX=5801H	リターン情報 AH=エラー・ステータス CX=データの項目数
機 能 マップ可能なセグメントの項目数を取得します。	

▶ INT67H(5900H) 拡張メモリのハードウェア情報の取得(OS/E 機能)	
入力パラメータ AX=5900H ES:DI=データ構造体	リターン情報 AH=エラー・ステータス
機 能 データ構造体の内容は次のとおりです。 1 ワード : Raw ページのサイズ 1 ワード : 代替レジスタ・セット 1 ワード : マッピング情報保存領域のサイズ 1 ワード : DMA レジスタ・セット 1 ワード : DMA チャネル操作	

▶ INT67H(5901H) Raw ページ数の取得(OS/E 機能)	
入力パラメータ AX=5901H	リターン情報 AH=エラー・ステータス BX=未使用 Raw ページ数 DX=総 Raw ページ数
機 能 非標準(16K バイト以外)のサイズの、ページ数を取得します。	

▶ INT67H(5A00H) 標準サイズのページの割り付け	
入力パラメータ AX=5A00H BX=ページ数	リターン情報 AH=エラー・ステータス DX=ハンドル
機 能 標準(16K バイト)のサイズのページを割り付けます。この機能ではページ 0 が割り付け可能です。	

▶ INT67H(5A01H) Raw ページの割り付け	
入力パラメータ AX=5A01H BX=ページ数	リターン情報 AH=エラー・ステータス DX=ハンドル
機 能 非標準(16K バイト以外)のサイズのページを割り付けます。この機能ではページ 0 が割り付け可能です。	

▶ INT67H(5B00H) 代替マップ・レジスタ・セットの取得(OS/E 機能)	
入力パラメータ AX=5B00H	リターン情報 AH=エラー・ステータス BL = 代替マップ・レジスタ番号 =0 の場合 ES:DI=マップ・レジスタの内容
機 能 代替マップ・レジスタ・セットを取得します。	

▶ INT67H(5B01H) 代替マップ・レジスタ・セットの設定(OS/E 機能)	
入力パラメータ AX=5B01H BL = 代替マップ・レジスタ番号 =0 の場合 ES:DI=マップ・レジスタの内容	リターン情報 AH=エラー・ステータス
機 能 代替マップ・レジスタ・セットの設定を行います。	

▶ INT67H(5B02H) 代替マップ・レジスタ保存領域のサイズ取得(OS/E 機能)	
入力パラメータ AX=5B02H	リターン情報 AH=エラー・ステータス DX = 保存領域のサイズ
機 能 代替マップ・レジスタ保存領域の、サイズを取得します。	

▶ INT67H(5B03H) 代替マップ・レジスタ・セットの割り付け(OS/E 機能)	
入力パラメータ AX=5B03H	リターン情報 AH=エラー・ステータス BL = レジスタ・セット番号 =0: サポートなし
機 能 代替マップ・レジスタ・セットの割り付けを行います。	

▶ INT67H(5B04H) 代替マップ・レジスタ・セットの開放(OS/E 機能)	
入力パラメータ AX=5B04H BL=レジスタ・セット番号	リターン情報 AH=エラー・ステータス
機 能 代替マップ・レジスタ・セットの開放を行います。	

▶ INT67H(5B05H) DMA マップ・レジスタ・セットの割り付け(OS/E 機能)	
入力パラメータ AX=5B05H	リターン情報 AH=エラー・ステータス BL=レジスタ・セット番号 =0: サポートなし
機 能 DMA マップ・レジスタ・セットの割り付けを行います。	

▶ INT67H(5B06H) DMA を有効にする(OS/E 機能)	
入力パラメータ AX=5B06H BL=レジスタ・セット番号 BL=チャンネル番号	リターン情報 AH=エラー・ステータス
機 能 DMA を有効にします。	

▶ INT67H(5B07H) DMA を無効にする(OS/E 機能)	
入力パラメータ AX=5B07H BL=レジスタ・セット番号	リターン情報 AH=エラー・ステータス
機 能 DMA を無効にします。	

▶ INT67H(5B08H) DMA マップ・レジスタ・セットの開放(OS/E 機能)	
入力パラメータ AX=5B08H BL = レジスタ・セット番号	リターン情報 AH=エラー・ステータス
機 能 DMA マップ・レジスタ・セットの開放を行います。	

▶ INT67H(5CH) ウォーム・ブートの準備	
入力パラメータ AH=5CH	リターン情報 AH=エラー・ステータス
機 能 ウォーム・ブートの準備を行います。	

▶ INT67H(5D00H) OS/E 機能を有効にする(OS/E 機能)	
入力パラメータ AH=5D00H BX : CX = アクセス・キー	リターン情報 AH=エラー・ステータス BX : CX = アクセス・キー
機 能 OS/E 機能を通常のアプリケーションが操作できないように、アクセス・キーをつけて有効にします。	

▶ INT67H(5D01H) OS/E 機能を無効にする(OS/E 機能)	
入力パラメータ AH=5D01H BX : CX = アクセス・キー	リターン情報 AH=エラー・ステータス BX : CX = アクセス・キー
機 能 OS/E 機能を無効にします。	

▶ INT67H(5D02H) アクセス・キーの返還(OS/E 機能)	
入力パラメータ AH=5D02H BX : CX = アクセス・キー	リターン情報 AH=エラー・ステータス BX : CX = アクセス・キー
機 能 アクセス・キーを EMM ドライバに返還し、初期状態にします。	

表 7-3 EMS のエラー・ステータス

ステータス	内 容
00H	正常終了
80H	EMM ドライバのソフトウェア・エラー
81H	EMS のハードウェア・エラー
82H	EMS メモリ管理ビジー
83H	ハンドルの誤り
84H	機能番号の指定の誤り
85H	使用可能なハンドルがない
86H	ページマップの復元エラー
87H	要求されたページ数が最大ページを超えている
88H	要求されたページ数が空いていない
89H	0 ページの割り付けは行えない
8AH	指定された論理ページはそのハンドルに割り当てられていない
8BH	物理ページ番号の指定に誤りがある
8CH	ページマップの保存領域がいっぱいになった
8DH	指定されたページ・マップはすでに保存されている
8EH	指定されたページ・マップはまだ保存されていない
8FH	副機能番号の指定の誤り
90H	属性の誤り
91H	システムは非破壊メモリのサポートをしていない
92H	転送元と転送先の領域が同一のハンドルで重なっている
93H	指定した領域の大きさがハンドルで指定されたものより大きい
94H	システム・メモリ領域と拡張メモリ領域が重なっている
95H	論理ページ内オフセットが論理ページの大きさを超えている
96H	領域の大きさが 1M バイトを超えている
97H	転送元と転送先の領域が同一ハンドルで重なっている
98H	転送元と転送先のメモリ・タイプの誤り
9AH	指定の代替マップ・レジスタ・セットがサポートされていない
9BH	すべての代替マップ DMA レジスタ・セットが使用されている
9CH	指定された代替マップ DMA レジスタ・セットが 0 でない
9DH	代替マップ DMA レジスタ・セットが定義されていない
9EH	専用の DMA チャンネルがサポートされていない
9FH	指定された DMA チャンネルがサポートされていない
A0H	指定したハンドル名に対するハンドル値がない
A1H	指定したハンドル名がすでに存在している
A2H	領域の転送・交換で 1M バイトを超えた
A3H	渡されたデータ構造が不正か破壊されている
A4H	現在この機能は使用できない

7.2 XMS(eXtended Memory Specification)

メモリ不足を解消する、とりあえずの方策として考案されたのがEMSならば、1Mバイトを超えるプロテクト・メモリを初めて正式にサポートしたものがXMSです。

DOS/VのXMSドライバは「HIMEM.SYS」のファイル名で提供されているため、今ひとつ知名度がありませんが、プロテクト・メモリにフォントをロードしなければならないDOS/Vにとっては、非常に重要なメモリ管理システムであるといえるでしょう。

表 7-4 XMS の機能一覧

機能番号	機能内容
00H	XMS バージョンの取得
01H	HMA の割り当て
02H	HMA の開放
03H	グローバルな A20 ラインの有効化
04H	グローバルな A20 ラインの無効化
05H	ローカルな A20 ラインの有効化
06H	ローカルな A20 ラインの無効化
07H	A20 ラインの状態取得
08H	EMB 使用可能領域サイズの取得
09H	EMB 領域の割り当て
0AH	EMB 領域の開放
0BH	EMB 領域のコピー
0CH	EMB 領域のロック
0DH	EMB 領域のロック解除
0EH	EMB のハンドル情報の取得
0FH	EMB 領域の再割り当て
10H	UMB 領域の割り当て
11H	UMB 領域の開放
12H	UMB 領域の再割り当て

XMS では次の 3 種のメモリ領域を管理します。

(1) UMB(Upper Memory Block)

640K バイトから 1M バイトの間の未使用の領域に、プロテクト・メモリの一部をマッピングした領域。見かけは 1M バイト以内の領域のため、プログラムの実行が可能なので、ドライバなどの常駐領域に使用することができる。

(2) HMA(High Memory Area)

セグメント FFFFH のオフセット 0010H から FFFFH までの約 64K バイト (64K-16 バイト) の領域。i8086CPU は 1M バイトまでしかメモリ管理ができないはずだったが、セグメントとオフセットの組み合わせ上のバグから偶然発生したエリアである。ただし 80286 以降の CPU しか使用できない。

DOS/V では DOS システムの一部をここに移動することで、本来のメイン・メモリを節約している。

(3) EMB(Extended Memory Block)

文字どおり 1M バイトより後半のプロテクト・メモリ。仮想 EMS ドライバは、実際にはこの領域を論理ページとしてマッピングして使用している。

XMS ではこの領域にはデータ領域としてしか利用できない。

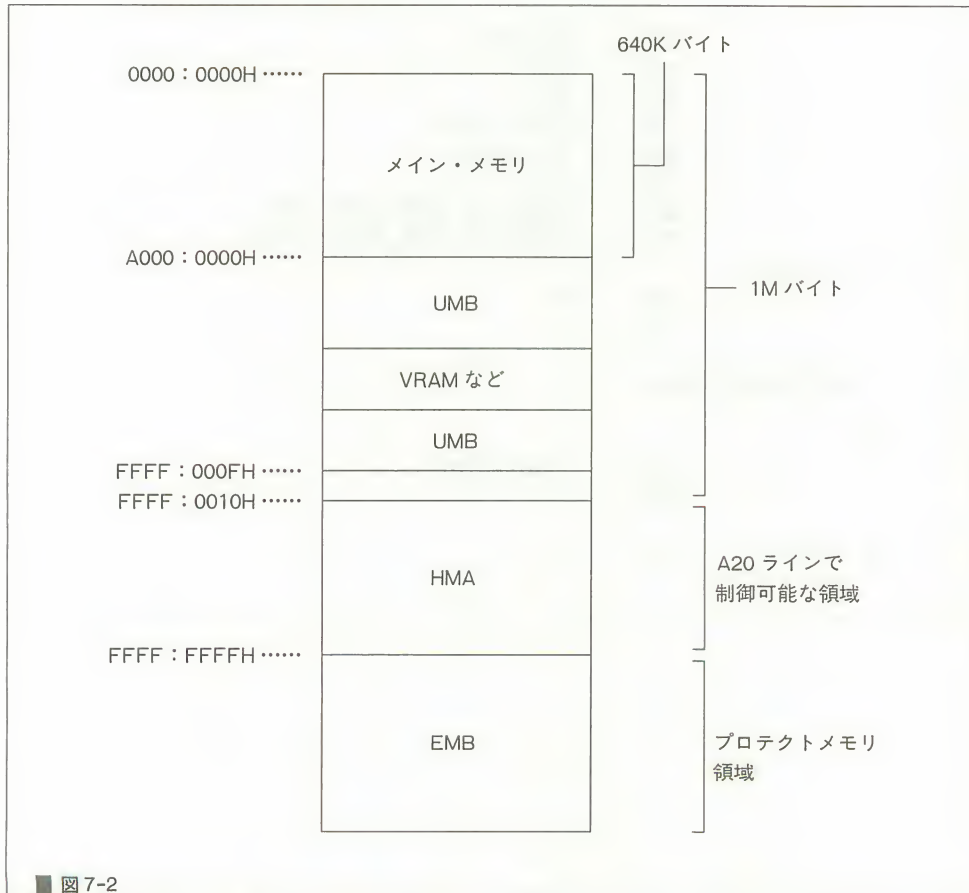


図 7-2

拡張メモリのイメージ

XMS の機能を利用するためには、多重割り込みの「XMS エントリ・アドレスの取得」(INT2FH, 4310H)でエントリ・アドレスを取得し、FAR コールします。

この際、事前に「XMS インストール・ステータスの取得」で XMS ドライバがロードされていることを確認しておかなければなりません。

XMS では、AX=0000H で BL のビット 7 が 1 の場合に、BL がエラー・ステータスとなります。

List 7-3

XMS 存在確認

```

;
; XMS存在確認とアドレス設定(存在しなければキャリーが立つ)
;
EXIST_XMS:
                PUSH    ES

                MOV     AX, 4300H
                INT     2FH
                SHL     AL, 1
                JB      EXIST_XMS_E
                MOV     AX, 4310H
                INT     2FH
                MOV     WORD PTR [XMS_ENTRY], BX
                MOV     WORD PTR [XMS_ENTRY+2], ES

EXIST_XMS_E:
                POP     ES
                RET

;
; XMS機能の呼び出し
;
XMS:
                CALL    [XMS_ENTRY]
                RET

XMS_ENTRY      DD      ?

```

以下、XMS の機能を各番号別に示します。

● HMA 関連の機能

▶ XMS(00H) XMS バージョンの取得	
入力パラメータ AH=00H	リターン情報 AX=XMS のバージョン BX=ドライバのバージョン DX=HMA 情報 0: HMA はない 1: HMA は存在する
機 能 XMS バージョンを取得します。	

▶ XMS(01H) HMA の割り当て	
入力パラメータ AH=01H DX=必要なバイト数	リターン情報 なし
機 能 HMA の最大値は 65520 バイトで、FFFF:0010H から始まっています。 HMA はセグメントが FFFFH の 1 つしか設定できない関係上、利用できるのは 1 つのプログラムからのみです。	

▶ XMS(02H) HMA の開放	
入力パラメータ AH=02H	リターン情報 なし
機 能 HMA の開放を行います。	

● A20 ライン関連の機能

▶ XMS(03H) グローバルな A20 ラインの有効化	
入力パラメータ AH=03H	リターン情報 なし
機 能 A20 ラインを有効にします。	

▶ XMS(04H) グローバルな A20 ラインの無効化	
入力パラメータ AH=04H	リターン情報 なし
機能 A20 ラインを有効にします。	

▶ XMS(05H) ローカルな A20 ラインの有効化	
入力パラメータ AH=05H	リターン情報 なし
機能 A20 ラインの制御カウンタを+1 し、A20 ラインを有効にします。	

▶ XMS(06H) ローカルな A20 ラインの無効化	
入力パラメータ AH=06H	リターン情報 なし
機能 A20 ラインの制御カウンタを-1 し、0 ならば A20 ラインを無効化します。したがって実際には、A20 ラインを有効に設定したすべてのプログラムが無効の設定をしないかぎり、無効にはなりません。	

▶ XMS(07H) A20 ラインの状態取得	
入力パラメータ AH=07H	リターン情報 AX=A20 ライン状態 0: 無効 1: 有効
機能 A20 ラインの状態を取得します。	

● EMB 関連の機能

▶ XMS(08H) EMB 使用可能領域サイズの取得	
入力パラメータ AH=08H	リターン情報 AX=最大サイズ DX=未使用サイズ
機能 取得される値は 1K バイト (1024 バイト) 単位です。	

▶ XMS(09H) EMB 領域の割り当て	
入力パラメータ AH=09H DX=必要なブロック数	リターン情報 AX=0001H DX=EMB ハンドル
機 能 ブロック・サイズは 1K バイト (1024 バイト) 単位です。	

▶ XMS(0AH) EMB 領域の開放	
入力パラメータ AH=0AH DX=EMB ハンドル	リターン情報 AX=0001H
機 能 EMB 領域の開放を行います。	

▶ XMS(OBH) EMB 領域のコピー

入力パラメータ

AH=0BH

DS:SI=データ構造体

リターン情報

AX=0001H

機能

設定するデータ構造体は次の構成です。

オフセット	サイズ	内 容
00H	2 ワード	EMB 領域の長さ(必ず偶数バイト)
04H	3 ワード	転送元のアドレス情報
0AH	3 ワード	転送先のアドレス情報

アドレス情報の内容は、指定する領域によって次のように変化します。

・コンベンショナル領域(メイン・メモリ)の場合

1 ワード : 0000H

1 ワード : 領域のオフセット

1 ワード : 領域のセグメント

・EMB 領域の場合

1 ワード : EMB ハンドル

2 ワード : 32 ビット・リニアアドレス

▶ XMS(0CH) EMB 領域のロック	
入力パラメータ AH=0CH DX=EMB ハンドル	リターン情報 AX=0001H DX:BX=ブロックの 32 ビット・アドレス
機 能 EMB 領域のロックを行います。	

▶ XMS(0DH) EMB 領域のロック解除	
入力パラメータ AH=0DH DX=EMB ハンドル	リターン情報 AX=0001H
機 能 EMB 領域のロック解除を行います。	

▶ XMS(0EH) EMB のハンドル情報の取得	
入力パラメータ AH=0EH DX=EMB ハンドル	リターン情報 AX=0001H BH=ロックしたブロック数 BL=利用可能なハンドル数 DX=ブロック・サイズ
機 能 ブロック・サイズは 1K バイト (1024 バイト) 単位です。	

▶ XMS(0FH) EMB 領域の再割り当て	
入力パラメータ AH=0FH BX=新しいブロック・サイズ DX=EMB ハンドル	リターン情報 AX=0001H
機 能 ブロック・サイズは 1K バイト (1024 バイト) 単位です。	

● UMB 関連の機能

▶ XMS(10H) UMB 領域の割り当て	
入力パラメータ AH=10H DX = 必要なブロック数	リターン情報 AX = 0001H BX = ブロックのセグメント DX = ブロックのパラグラフサイズ
機 能 ブロック・サイズは 1K バイト (1024 バイト) 単位です。	

▶ XMS(11H) UMB 領域の開放	
入力パラメータ AH=11H DX = ブロックのセグメント	リターン情報 AX = 0001H
機 能 UMB 領域を開放します。	

▶ XMS(12H) UMB 領域の再割り当て	
入力パラメータ AH=12H BX = 新しいブロック・サイズ (パラグラフ) DX = ブロックのセグメント	リターン情報 AX = 0001H
機 能 ブロック・サイズは 1K バイト (1024 バイト) 単位です。	

表 7-5 XMS のエラー・ステータス

ステータス	内 容
80H	関数がインプリメントされていない
81H	VDISK にデバイス・エラーが見つかった
82H	A20 にエラーが発生した
8EH	一般的なドライバ・エラー
8FH	修復不能なドライバ・エラーが発生
90H	HMA が存在しない
91H	HMA がすでに使用されている
92H	DX が/HMAMIN で指定された値より少ない
93H	HMA が割り当てられていない
94H	A20 ラインがまだ使用可能状態ではない
A0H	すべての拡張メモリが割り当てられた
A1H	EMM ハンドルがすべて使い果たされた
A2H	ハンドルが無効
A3H	転送元ハンドルが無効
A4H	転送元オフセットが無効
A5H	転送先ハンドルが無効
A6H	転送先オフセットが無効
A7H	転送サイズが無効
A8H	移動要求において 1M バイトを超えた
A9H	パリティ・エラーが見つかった
AAH	ブロックがロックされていない
ABH	ブロックがロックされている
ACH	ロックカウントがオーバーフローした
ADH	ロックが失敗した
B0H	スモール UMB が使用可能
B1H	UMB が利用できない
B2H	UMB のセグメント番号が無効

第 8 章

ハードウェア

OADGの規定によれば、DOS/Vが対応するハードウェアには、次のものがあげられています。

- ・ IBM PC/AT (VGA 登載)
- ・ IBM PS/2 (AT バスモデル)
- ・ 上記機種 of 互換機

OADGのガイドラインにも、「OADGが目指すハードウェア・インタフェースの基本となるのは、国際的に広く使用されている IBM-PC/AT およびディスプレイ制御モジュールの VGA です」と明言されています。

したがって、ハードウェアは IBM-PC/AT が事実上の基準となるわけですが、現実問題として、IBM-PC 互換機では、オリジナルの IBM-PC と若干異なるハードウェアを使用している場合も少なくありません。

IBM-PC では BIOS が充実しているため、これらの相違は BIOS に吸収させることで、ハードウェアを強化した互換機も存在するのです(そのため、同じ PC 互換機でも BIOS, ROM を差し替えたりすると動作しない場合が多い)。

DOS/V はあくまで「ソフトウェアで日本語環境を実現」したシステムですから、基本的にハードウェアを直接制御することは避けなければなりません。

しかし、ドライバやゲーム・プログラムなど、どうしてもハードウェアの直接制御を必要とする場合があるのも事実です。

本書ではこれらの理由から、DOS/V のアプリケーション開発の都合上どうしても触れざるをえないハードウェアにのみ解説を加え、それ以外のハードウェアは概要を取り上げるにとどめました。

表 8-1 システム I/O アドレスの使用状況の概要

システム・ボード	
000H ~ 01FH	DMA コントローラ 1 (i8237A)
020H ~ 03FH	割り込みコントローラ 1 (i8259A)
040H ~ 05FH	システム・タイマ (i8254)
060H ~ 06FH	キーボード (i8042)
070H ~ 07FH	リアルタイム・クロックと CMOS-RAM (MC146818)
080H ~ 08FH	DMA ページ・レジスタ (74LS612)
090H ~ 09FH	システム・ボード・コントローラ (PS/2 のみ)
0A0H ~ 0AFH	割り込みコントローラ 2 (i8259A)
0C0H ~ 0DFH	DMA コントローラ 2 (i8237A)
0F0H ~ 0FFH	数値演算コプロセッサ (i80 ? 87)

I/O チャンネル	
1F0H ~ 1F8H	ハードディスク・コントローラ
2F8H ~ 2FFH	シリアル・ポート 2 (i8251A)
3F0H ~ 3F7H	フロッピーディスク・コントローラ (μ PD765A)
3F8H ~ 3FFH	シリアル・ポート 1 (i8251A)

※ パラレル・ポートは機種ごとに異なっている

ビデオ・ボード (VGA 対応範囲)	
3B0H ~ 3BFH	MDA
3C0H ~ 3CFH	EGA/VGA
3D0H ~ 3DFH	CGA

※ ()内は標準的に使用される LSI 名

8.1 割り込みコントローラ(i8259A)

割り込みコントローラは i8259A 相当品が 2 個使用され、カスケード接続されています。マスタは PIC #1 (I/O アドレス 20H, 21H)、スレーブは PIC #2 (I/O アドレス A0H, A1H) で、PIC #2 からの割り込み要求は PIC #1 の IRQ2 に入力されています。

割り込みベクタの設定や取得は、直接メモリを操作せず、必ずファンクション・コールの

「割り込みベクタの設定」(INT21H, AH=25H)

「割り込みベクタの取得」(INT21H, AH=35H)

を使用するようにします。

表 8-2 割り込みレベル

レベル	INT ベクタ	内 容
NMI		
	INT02H	メモリ・パリティ・エラー
マスタ・コントローラ		
IRQ0	INT08H	システム・タイマ
IRQ1	INT09H	キーボード
IRQ2	INT0AH	スレーブへのカスケード
IRQ3	INT0BH	シリアル 2
IRQ4	INT0CH	シリアル 1
IRQ5	INT0DH	パラレル 2
IRQ6	INT0EH	フロッピーディスク
IRQ7	INT0FH	パラレル 1
スレーブ・コントローラ		
IRQ8	INT70H	リアルタイム・クロック
IRQ9	INT71H	マスタからのカスケード
IRQA	INT72H	予約済み
IRQB	INT73H	予約済み
IRQC	INT74H	(マウス)
IRQD	INT75H	数値演算コプロセッサ
IRQE	INT76H	ハードディスク
IRQF	INT77H	予約済み

以下に、I/O ポートアドレスを示します。

● PIC # 1(マスタ)

アドレス	R/W	機 能
20H	R	IRR：割り込み要求レジスタ ISR：インサース・レジスタ
	W	ICW1：イニシャライズ・コマンド・ワード OCW2・3：オペレーション・コマンド・ワード
21H	R	IMR：インタラプト・マスク・レジスタ
	W	OCW1：オペレーション・コマンド・ワード ICW2・3・4：イニシャライズ・コマンド・ワード

● PIC # 2(スレーブ)

アドレス	R/W	機 能
A0H	R	IRR：割り込み要求レジスタ ISR：インサース・レジスタ
	W	ICW1：イニシャライズ・コマンド・ワード OCW2・3：オペレーション・コマンド・ワード
A1H	R	IMR：インタラプト・マスク・レジスタ
	W	OCW1：オペレーション・コマンド・ワード ICW2・3・4：イニシャライズ・コマンド・ワード

(1) イニシャライズ・コマンド・ワード

[ICW1(ライト)]

	D7	D6	D5	D4	D3	D2	D1	D0
20H A0H	0	0	0	1	TRGM	0	SNGL	ICW4

TRGM=1：レベル・トリガ入力モード

0：エッジ・トリガ入力モード(デフォルト)

SNGL=1：シングル接続

0：カスケード接続(デフォルト)

ICW4 = 1：ICW4 が必要

0：ICW4 は不要

[ICW2(ライト)]

	D7	D6	D5	D4	D3	D2	D1	D0
2IH AIH	IBVC					IRQL		

IBVC=割り込みベース・ベクタ

デフォルト: PIC # 1=08H

: PIC # 2=070H

IRQL=割り込みリクエスト・レベル

[ICW3(ライト)]

・マスタ(PIC # 1)

	D7	D6	D5	D4	D3	D2	D1	D0
2IH	S7	S6	S5	S4	S3	S2	S1	S0

S7 ~ S0=各 IRQ のスレープ入力フラグ(デフォルト: 04H)

1 で IRQ はスレープをもつ

・スレープ(PIC # 2)

	D7	D6	D5	D4	D3	D2	D1	D0
AIH	0	0	0	0	0	SLID		

SLID=スレープ ID コード(デフォルトは 02H)

マスタの何番に IRQ 接続されているかを設定

[ICW4(ライト)]

	D7	D6	D5	D4	D3	D2	D1	D0
2IH AIH	0	0	0	SFNM	BUFM	0	0	1

SFNM=1: スペシャル・フリーネステッド・モード

0: 非スペシャル・フリーネステッド・モード(デフォルト)

BUFM=1: バッファ・モード

0: 非バッファ・モード(デフォルト)

(2) オペレーション・コマンド・ワード

[OCW1(ライト)・IMR(リード)]

	D7	D6	D5	D4	D3	D2	D1	D0
21H A1H	M7	M6	M5	M4	M3	M2	M1	M0

M7 ~ M0 = 各 IRQ の割り込みマスク

1 : 割り込み禁止

0 : 割り込み許可

[OCW2(ライト)]

	D7	D6	D5	D4	D3	D2	D1	D0
20H A0H	R	SL	EOI	0	0	SLEV		

R	SL	EOI	機 能	
0	0	1	非特殊 EOI コマンド	割り込み終了
0	1	1	特殊 EOI コマンド *	
1	0	1	非特殊 EOI コマンドで回転	自動回転
1	0	0	自動 EOI モードで回転(セット)	
0	0	0	自動 EOI モードで回転(クリア)	
1	1	1	特殊 EOI コマンドで回転 *	特殊回転
1	1	0	優先セット・コマンド *	
0	1	0	何もしない	

* 印の場合は、IRQ のレベルを SLEV に設定

[OCW3(ライト)]

	D7	D6	D5	D4	D3	D2	D1	D0
20H A0H	0	SMM		0	1	PC	RRS	

SMM = 0 または 1 : 何もしない

2 : スペシャル・マスクのクリア

3 : スペシャル・マスクのセット

PC = 1 : ポール・コマンド

0 : 非ポール・コマンド(デフォルト)

RRS = 0 または 1 : 何もしない

2 : IRR 読み込み

3 : ISR 読み込み

List 8-1

一般的なマスタ割り込みの EOI

INTRUPT:

```
CLI  
PUSH    AX
```

割り込み処理

; マスタPICへのEOI

```
MOV     AL, 20H  
OUT     (20H), AL
```

```
POP     AX  
STI  
IRET
```

8.2 DMA コントローラ (i8237A)

DMA コントローラは、i8237A 相当品が 2 個使用されています。DMAC #1 は 8 ビットの I/O とメモリ間転送を、DMAC #2 は 16 ビットの I/O とメモリ間転送を制御しています。

また、従来の DMAC の機能では転送アドレスが 64K バイトしかなかったため、これを補うためにページ・レジスタが増設されています。

DMAC はメモリ・メモリ間転送はできないため、実際にはディスク制御程度にしか利用されていません。アプリケーション側から利用する機会もないと思われるので、ここでは I/O ポートの概要を取り上げるにとどめます。

表 8-3 DMA の利用状況

DMAC #1	チャンネル 0	未使用
	チャンネル 1	未使用
	チャンネル 2	フロッピーディスク
	チャンネル 3	ハードディスク
DMAC #2	チャンネル 4	DMAC #1 カスケード
	チャンネル 5	未使用
	チャンネル 6	未使用
	チャンネル 7	未使用

以下に、I/O ポート・アドレスを示します (ポート上段：DMAC #1, 下段：DMAC #2)。

ポート	R/W	機 能	
00H	W	チャンネル 0	ベース & カレント・アドレス
C0H	R	チャンネル 4	カレント・アドレス
01H	W	チャンネル 4	ベース & カレント・ワード・カウンタ
C2H	R		カレント・ワード・カウンタ
02H	W	チャンネル 1	ベース & カレント・アドレス
C4H	R	チャンネル 5	カレント・アドレス
03H	W	チャンネル 5	ベース & カレント・ワード・カウンタ
C6H	R		カレント・ワード・カウンタ
04H	W	チャンネル 2	ベース & カレント・アドレス
C8H	R	チャンネル 6	カレント・アドレス
05H	W	チャンネル 6	ベース & カレント・ワード・カウンタ
CAH	R		カレント・ワード・カウンタ

06H	W	チャンネル 3 チャンネル 7	ベース & カレント・アドレス
CCH	R		カレント・アドレス
07H	W		ベース & カレント・ワード・カウンタ
CEH	R		カレント・ワード・カウンタ
08H	W	コマンド・レジスタ	
D0H	R	ステータス・レジスタ	
09H	W	リクエスト・レジスタ	
D2H			
0AH	W	シングル・マスク・レジスタ・ビット	
D4H			
0BH	W	モード・レジスタ	
D6H			
0CH	W	バイト・ポインタ・フリップフロップ	
D8H			
0DH	W	マスタ・クリア	
DAH	R	テンポラリ・レジスタ	
0EH	W	マスク・レジスタ・クリア	
DCH			
0FH	W	オール・マスク・レジスタ	
DEH			

● DMAC # 1 ページ・レジスタ

ポート	R/W	機 能
81H	W	チャンネル 2
82H	W	チャンネル 3
83H	W	チャンネル 1
87H	W	チャンネル 0

● DMAC # 2 ページ・レジスタ

ポート	R/W	機 能
89H	W	チャンネル 6
8AH	W	チャンネル 7
8BH	W	チャンネル 5
8FH	W	チャンネル 4

8.3 システム・タイマ(i8254)

システム・タイマは i8254 相当品が使用されており、3 個のプログラマブル・カウンタを内蔵しています。

クロックは 1.19318MHz でドライブされています。

以下に、I/O ポート・アドレスを示します。

ポート	R/W	機 能	モード
40H	R/W	カウンタ#0(インターバル・タイマ)	3
41H		カウンタ#1(DRAM リフレッシュ)	2
42H		カウンタ#2(スピーカー)	3
43H	W	モード・レジスタ	

(1) モード・レジスタ(ライト)

[カウンタ選択]

	D7	D6	D5	D4	D3	D2	D1	D0
43H	CHNO		RWMD		MODE		BCD	

CHNO = 00 : カウンタ#0 選択

01 : カウンタ#1 選択

10 : カウンタ#2 選択

11 : リードバック・コマンド

RWMD = 00 : カウンタ・ラッチ動作(読み出し時)

01 : 下位バイトの読み書き

10 : 上位バイトの読み書き

11 : 下位・上位の順に読み書き(デフォルト)

MODE = モード番号

000 : カウント終了時の割り込み

010 : レート・ジェネレータ

011 : 方形波レート・ジェネレータ

BCD = 1 : BCD カウント

0 : バイナリ・カウント

[リードバック・コマンド]

	D7	D6	D5	D4	D3	D2	D1	D0
43H	1	1	CR	SR	CH2	CH1	CH0	0

CR = 0 : カウンタ・ラッチ
 SR = 0 : ステータス・ラッチ
 CH2 = 1 でカウンタ#2 選択
 CH1 = 1 でカウンタ#1 選択
 CH0 = 1 でカウンタ#0 選択

(2) ステータス(リード)

	D7	D6	D5	D4	D3	D2	D1	D0
40H	OUT	NULL	RWMD			MODE		BCD
41H								
42H								

OUT = 1 : OUT 端子レベル・ハイ
 0 : OUT 端子レベル・ロー
 NULL = 0 : カウンタ有効
 1 : カウンタ無効
 RWMD = 00 : カウンタ・ラッチ動作
 01 : 下位バイトの読み書き
 10 : 上位バイトの読み書き
 11 : 下位・上位の順に読み書き(デフォルト)
 MODE = モード番号
 000 : カウント終了時の割り込み
 010 : レート・ジェネレータ
 011 : 方形波レート・ジェネレータ
 BCD = 1 : BCD カウント
 0 : バイナリ・カウント

(3) カウンタの読み書き(リード/ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
40H	COUNTER							
41H								
42H								

※ モード・レジスタの設定直後に有効

List 8-2

ビープ音による音程発生

```

;
; BX = 周波数(Hz)
;
; 参考:
;
;          カウンタ   =    $\frac{1, 193, 180}{\text{周波数}}$ 
;
SOUND:
    CLI
                                ; モード設定
    MOV     AL, 10110110B
    OUT     (43H), AL

    MOV     AX, WORD PTR [CLOCK]
    MOV     DX, WORD PTR [CLOCK+2]
    DIV     BX
                                ; カウンタ計算

    OUT     (42H), AL
    OUT     (42H), AH
                                ; カウンタ設定

    STI
                                ; ビープON

    IN      AL, (61H)
    AND     AL, 11111110B
    OR      AL, 00000001B
    OUT     (61H), AL

    適当なウェイト
                                ; ビープOFF

    IN      AL, (61H)
    AND     AL, 11111110B
    OUT     (61H), AL

    RET

CLOCK      DD      1193180
                                ; 基準クロック

```

8.4 リアルタイム・クロックと CMOS-RAM(MC146818)

リアルタイム・クロックは MC146818 相当品が使用されています。リアルタイム・クロックは内部に 64 バイトの CMOS-RAM をもっていますが、これはバッテリー・バックアップされているので、時間や各種情報を保存するために使用されています。

以下に、I/O ポート・アドレスを示します。

ポート	R/W	機 能
70H	W	CMOS-RAM のアドレス指定
71H	R/W	CMOS-RAM データの読み書き

(1) CMOS-RAM のアドレス指定(ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
70H	MASK	0	CMAD					

MASK = 1 : NMI 禁止

0 : NMI 許可

CMAD = CMOS-RAM アドレス

表 8-4 CMOS-RAM の内容

オフセット	サイズ	内 容
カレンダー情報		
00H	1 バイト	秒
01H	1 バイト	アラームの秒
02H	1 バイト	分
03H	1 バイト	アラームの分
04H	1 バイト	時
05H	1 バイト	アラームの時
06H	1 バイト	曜日
07H	1 バイト	日
08H	1 バイト	月
09H	1 バイト	年

リアルタイム・クロック・ステータス		
0AH	1 バイト	ステータス・レジスタ A ビット 7 : 時刻更新サイクル ビット 6 ~ 4 : クロック分周期 ビット 3 ~ 0 : 割り込み周期
0BH	1 バイト	ステータス・レジスタ B ビット 7 : 動作状態 0 = 動作中, 1 = 停止中 ビット 6 : 周期割り込み 1 = 許可, 0 = 禁止 ビット 5 : アラーム割り込み 1 = 許可, 0 = 禁止 ビット 4 : 時刻更新サイクル終了割り込み 1 = 許可, 0 = 禁止 ビット 2 : カレンダ情報 1 = バイナリ・モード 0 = BCD モード (デフォルト) ビット 1 : 1 = 24 時間モード (デフォルト) 0 = 12 時間モード
0CH	1 バイト	ステータス・レジスタ C ビット 7 : IRQ 要求フラグ ビット 6 : 周期割り込みフラグ ビット 5 : アラーム割り込みフラグ ビット 4 : 時間更新終了割り込みフラグ
0DH	1 バイト	ステータス・レジスタ D ビット 7 : リアルタイム・クロックの動作 1 = 情報は有効 0 = バッテリ異常
BIOS ワークエリア		
0EH	1 バイト	ビット 7 : RTC 電池切れ ビット 6 : CMOS-RAM チェックサム・エラー ビット 5 : セルフテストで構成が違う ビット 4 : セルフテストでメモリサイズが違う ビット 3 : ハードディスクの初期化に失敗 ビット 2 : RTC の時刻がおかしい
0FH	1 バイト	リセット要因 0 : パワーオン・リセット 1 : リアルモードへの移行 4 : セルフテストの終了とリブート 5 : EOI 発行後, [40:67H] へ JUMP 8 : メモリ・セルフテストへの復帰 9 : INT15H, AH=87H への復帰 10 : [40:67H] へ JUMP 11 : [40:67H] へ IRET 12 : [40:67H] へ RET

10H	1 バイト	ビット 7～4：フロッピー・ドライブ 0 のタイプ ビット 3～0：フロッピー・ドライブ 1 のタイプ 0000＝なし 0001＝360K バイト 0010＝1.2M バイト 0011＝720K バイト 0100＝1.44M バイト
11H	1 バイト	予約済み
12H	1 バイト	ビット 7～4：ハードディスク 0 のタイプ ビット 3～0：ハードディスク 1 のタイプ 1111＝19H と 1AH 参照 (機種に依存)
13H	1 バイト	予約済み
14H	1 バイト	ビット 7～6：フロッピー・ドライブの台数 00＝1 ドライブ 01＝2 ドライブ ビット 5～4：ディスプレイ形式 00：ROM 登載カード 01：40 桁カラー 10：80 桁カラー 11：モノクロ ビット 1：数値演算コプロセッサ登載 ビット 0：フロッピーからブート可能
15H	1 ワード	リアルモード RAM 容量(K バイト単位)
17H	1 ワード	拡張 RAM 容量(K バイト単位)
19H	1 バイト	ハードディスク 0 の拡張タイプ
1AH	1 バイト	ハードディスク 1 の拡張タイプ
1BH ↓	19 バイト	予約済み
2EH	1 バイト	CMOS-RAM チェックサムの上位
2FH	1 バイト	CMOS-RAM チェックサムの下位
30H	1 ワード	拡張 RAM 容量
32H ↓ 3FH	14 バイト	予約済み

※ PS/2 では、32H と 33H には 10H から 31H までの CRC データが入る
(32H：上位バイト，33H：下位バイト)

8.5 システム・ポート

以下に、I/O ポート・アドレスを示します。

ポート	R/W	機 能
61H	R	システム・ステータス
	W	システム・コマンド

(1) システム・ステータス(リード)

	D7	D6	D5	D4	D3	D2	D1	D0
61H	MPE	IOCE	TC20	REF	EIOC	EMPE	SPKG	T2G

MPE = メモリ・パリティ・エラー

IOCE = I/O チャンネル・エラー

TC20 = チャンネル 2 の出力信号がアクティブ

REF = メモリ・リフレッシュのチェック

EIOC = I/O チャンネル・エラー状態

EMPE = メモリ・パリティ・チェック状態

SPKG = PIT チャンネル 2 のスピーカへの出力状態

T2G = PIT チャンネル 2 の出力状態

(2) システム・コマンド(ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
61H	0	0	0	0	EIOC	EMPE	SPKG	T2G

EIOC = I/O チャンネル・エラー状態

1: 禁止, 0: 許可

EMPE = メモリ・パリティ・チェック状態

1: 禁止, 0: 許可

SPKG = PIT チャンネル 2 のスピーカへの出力状態

1: 禁止, 0: 許可

T2G = PIT チャンネル 2 の出力状態

1: 禁止, 0: 許可

8.6 キーボード(i8042)

キーボードはハード的に外部のデバイスなので、コントローラに i8042 相当品を使用し、通信でインタフェースを行っています。i8042 は内部に ROM をもった 1 チップ・マイコンで、キーボード・コントローラとしてプログラムされています。

DOS/V では、複数のキーボードへの対応が求められているため、基本的にはキーボード BIOS を使用すべきで、ハードウェアを直接操作すべきではありません。

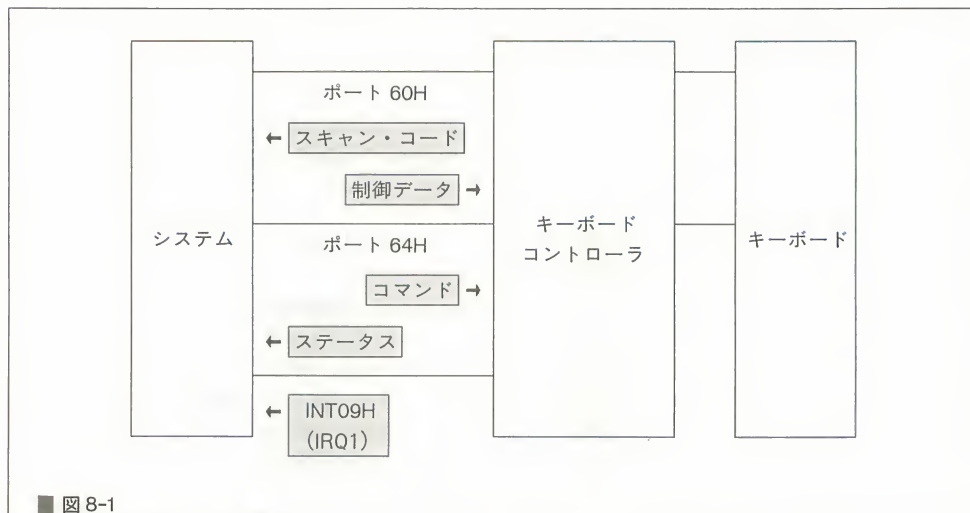


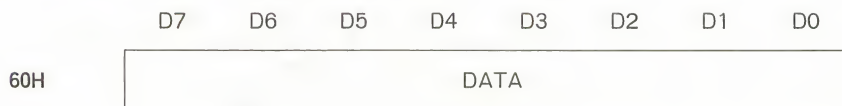
図 8-1

キーボード・システム構成

以下に、I/O ポート・アドレスを示します。

ポート	R/W	機 能
60H	R/W	キーボード・データ
	W	キーボード・コマンド
64H	R	キーボード・ステータス

(1) キーボード・データ(リード/ライト)



DATA = キーボード・データ

(2) キーボード・コマンド(ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
64H	COM							

COM=キーボード・コマンド

(3) キーボード・ステータス(リード)

	D7	D6	D5	D4	D3	D2	D1	D0
64H	PE	RDT0	TDT0	IBSW	CDWS	SYSF	FINP	FOUT

PE = 1: パリティ・エラー

RDT0 = 1: 受信タイムアウト

TDT0 = 1: 送信タイムアウト

IBSW = 0: インhibit状態

CDWS = 0: データ・ポート (60H) にライトした

1: コマンド・ポート (64H) にライトした

SYSF = 0: パワー ON リセット後

FINP = 1: 入力バッファがいっぱい

FOUT = 1: 出力バッファがいっぱい

表 8-5 キーボードへのコマンド

コマンド	機 能
EDH	セット/リセット・ステータス・インジケータ
EEH	エコー
F0H	代替走査コード選択
F2H	ID 読み出し
F3H	セット・タイプマティック・レート/ディレイ
F4H	イネーブル
F5H	デフォルト・ディセーブル
F6H	セット・デフォルト
F7H	セット・オール・キー(タイプマティック)
F8H	セット・オール・キー(メイク/ブレイク)
F9H	セット・オール・キー(メイク)
FAH	セット・オール・キー(タイプマティック/メイク/ブレイク)
FBH	セット・キー・タイプ(タイプマティック)
FCH	セット・キー・タイプ(メイク/ブレイク)
FDH	セット・キー・タイプ(メイク)
FEH	再送信
FFH	リセット

表 8-6 キーボードからのコマンド

コマンド	機 能
00H	オーバーラン
83H・ABH	キーボード ID
AAH	BAT 完了コード
EEH	エコー
F0H	ブレーク・コード・プリフィクス
FAH	アクノリッジ応答
FCH	BAT 障害コード
FEH	再送信
FFH	オーバーラン

8.7 ビデオ(VGA)

DOS/V のビデオ・ハードウェアは、その名が示すとおり VGA です。

VGA = Video Graphic Array

VGA はこれまでの CGA, EGA などのビデオ・ボードの機能をカバーし、高度な論理演算機能や多画面同時アクセス機能などの、高速な描画機能を提供しています。

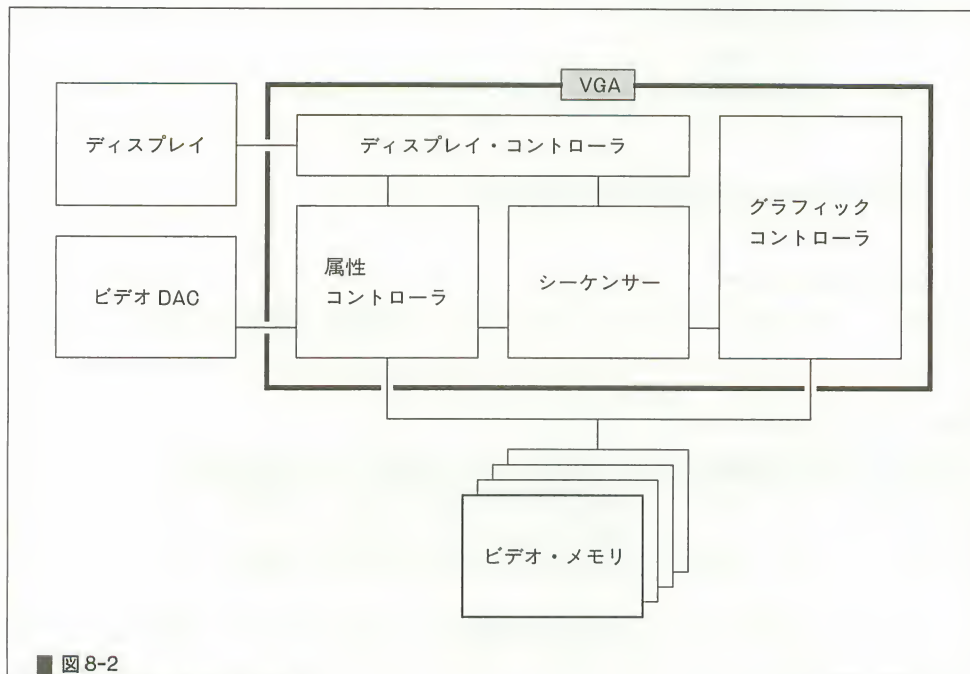


図 8-2

VGA ビデオ・サブ・システム

VGA は極めて多彩なモードをサポートしていますが、DOS/V を使っている限りは、英語モードでのアプリケーション開発はあまり意味をもたないので、本書ではアプリケーションで利用する可能性の高い、いくつかのモードに限定して解説を行います。

VGA のコントローラの設定方法にはいくつかの種類がありますが、最も一般的なのがインデックス方式です。これは1つのポートに対してインデックス(レジスタ番号指定)を指定することで、少ないポート数で多くのレジスタ制御を行う方法です。

たとえばシーケンサーの「レジスタ 01H」に「データ 02H」を設定したい場合は、次の2つの設定方法があります。

8ビットで設定する方法	
MOV	AL, 01H
OUT	(3C4H), AL
MOV	AL, 02H
OUT	(3C5H), AL

16ビットで設定する方法	
MOV	AX, 0201H
MOV	DX, 3C4H
OUT	(DX), AX

※ AL の値は 3C4H, AH の値は 3C5H に出力される

(1) テキスト・モード

英語モード時の次のビデオ・モードは、テキスト VRAM を使用したテキスト・モードです。

表 8-7 VGA テキスト・モード

ビデオ・モード	互換ポート	サイズ	色数	セグメント
00H・01H	CGA	40 桁×25 行	カラー 16 色	B800H
02H・03H		80 桁×25 行	カラー 16 色	
07H	MDA	80 桁×25 行	モノクロ 2 色	B000H

いずれのモードでもテキスト VRAM の偶数アドレスが文字コード、奇数アドレスが属性となっています。

表 8-8 カラーモードのアトリビュート

ビット 7	背景色輝度またはブリック		
ビット 6～4	背景色		
ビット 3	文字色輝度またはキャラクタセット		
ビット 2～0	文字色		
背景・文字の色定義			
0：黒	4：赤	8：灰色	C：薄い赤
1：青	5：紫	9：薄い青	D：薄い紫
2：緑	6：茶	A：薄い緑	E：黄色
3：水色	7：白	B：薄い水色	F：明るい白

表 8-9 モノクロモードのアトリビュート

ビット 7	ブリンク
ビット 6 ~ 0	0000000 : 黒 0000001 : アンダーライン 0000111 : 通常表示 0001111 : 高輝度 0001001 : 高輝度 + アンダーライン 1110000 : 反転

(2) グラフィック・モード

グラフィック・モードには次の英語モード時のビデオ・モードが対応しています。

表 8-10 VGA のグラフィック・モード

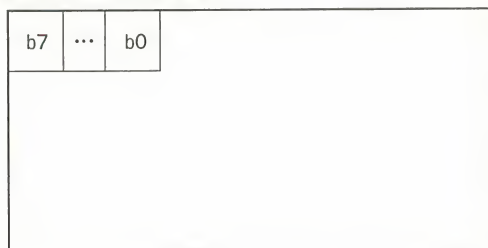
モード	互換ボード	解像度	色数	セグメント	ページ数
04H	CGA	320×200	4	B800H	1
05H		320×200	4	B800H	1
06H		640×200	2	B800H	1
0DH	EGA	320×200	16	A000H	8
0EH		640×200	16	A000H	4
0FH		640×350	2	A000H	2
10H		640×350	16	A000H	2
11H	VGA	640×480	2	A000H	1
12H		640×480	16	A000H	1
13H		320×200	256	A000H	1

表 8-11 各グラフィック・モード時の基本色

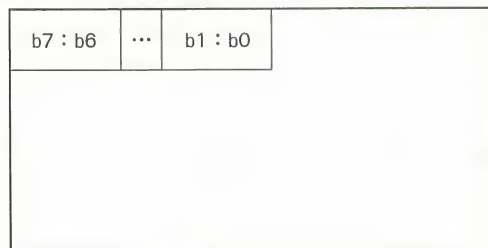
モード	設定色(値は 16 進数)			
04H	0 : 黒	1 : 薄い水色	2 : 薄い紫	3 : 明るい白
05H	0 : 黒	1 : 緑	2 : 赤	3 : 茶
06H	0 : 黒	1 : 明るい白		
0FH	0 : 黒	1 : 白	2 : 点滅白	3 : 明るい白
0DH	0 : 黒	1 : 青	2 : 緑	3 : 水色
0EH	4 : 赤	5 : 紫	6 : 茶	7 : 白
10H	8 : 灰色	9 : 薄い青	A : 薄い緑	B : 薄い水色
12H	C : 薄い赤	D : 薄い紫	E : 黄色	F : 明るい白
13H	0 ~ F : 上記と同様 10 ~ 1F : グラデーション 20 ~ FF : 汎用色セット			

※ これらの初期値は BIOS によって設定される

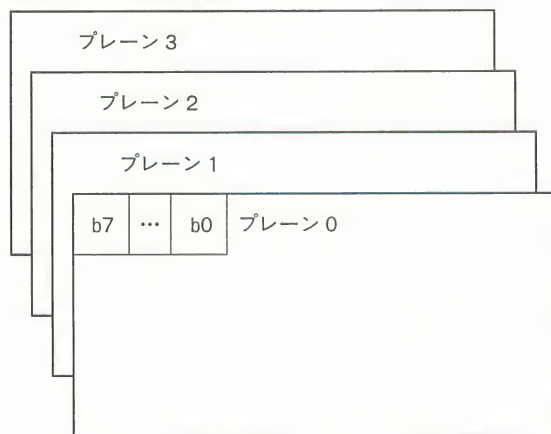
・ 2 色モードの VRAM 構造



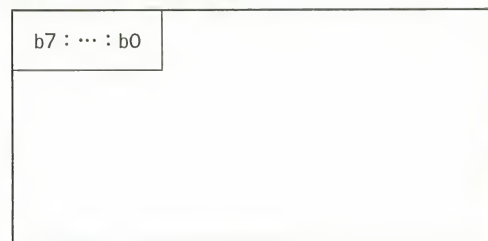
・ 4 色モードの VRAM 構造



・ 16 色モードの VRAM 構造



・ 256 色モードの VRAM 構造




※  は 1 ドット

図 8-3
各グラフィック・モード時の VRAM 構造

(3) 汎用レジスタ

以下に、汎用レジスタ I/O ポート・アドレスを示します。

読み込み ポート	書き込み ポート	機 能
3CCH	3C2H	多目的出力レジスタ
3C2H	----	入カステータス・レジスタ 0
3BAH 3DAH	----	入カステータス・レジスタ 1 (モノクロ用) (カラー用)
3CAH	3BAH 3CAH	機能制御レジスタ (モノクロ用) (カラー用)
3C3H	3C3H	VGA イネーブル・レジスタ

[多目的出力レジスタ]

D7	D6	D5	D4	D3	D2	D1	D0
VLIN	PGSL	0	CLSL	ERAM	IOSL		

VLIN = 00 : 予約済み

01 : 400 ライン

10 : 350 ライン

11 : 480 ライン

PGSL = 0 : 低位 64KB のメモリ・ページ選択

1 : 高位 64KB のメモリ・ページ選択

CLSL = 00 : 横 640 ドット・モード

01 : 横 720 ドット・モード

10 : 予約済み

11 : 予約済み

ERAM = 1 : ビデオ RAM へのアクセス許可

IOSL = 1 : カラー・モード

0 : モノクロ・モード

[入カステータス・レジスタ 0]

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	DSPM	0	0	0	0

DSPM = 0 : モノクロ・ディスプレイ接続

1 : カラー・ディスプレイ接続

[入カステータス・レジスタ I]

D7	D6	D5	D4	D3	D2	D1	D0
0	0	DIAG	VRET	0	0	DSPE	

DIAG = 診断用ビット
 VRET = 1 : 垂直帰線中
 DSPE = 0 : 表示期間中

[機能制御レジスタ]

D7	D6	D5	D4	D3	D2	D1	D0
FC1				0	FC2		

FC1 = 予約済み
 FC2 = 予約済み

[VGA イネーブル・レジスタ]

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	VGAE

VGAE = 1 : VGA イネーブル

(4) シーケンサー

以下に、シーケンサーのレジスタを示します。

(インデックス = 3C4H, データ = 3C5H)

INDX	R/W	機 能
00H	R/W	リセット・レジスタ
01H	R/W	クロッキング・モード・レジスタ
02H	R/W	マップ・マスク・レジスタ
03H	R/W	文字マップ選択レジスタ
04H	R/W	メモリ・モード・レジスタ

[リセット・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
00H	0	0	0	0	0	0	SRST	ARST

SRST=0: 同期リセット

1: シーケンサー動作

ARST=0: 非同期リセット

1: シーケンサー動作

[クロッキング・モード・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
01H	0	0	SOFF	AD32	CLK2	AD16	0	CWID

SOFF=1: 画面オフ

AD32=1: 32ビット・アドレス指定

CLK2=1: クロック 2 分周指定

AD16=1: 16ビット・アドレス指定

CWID=0: 文字幅 9 ドット

1: 文字幅 8 ドット

[マップ・マスク・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
02H	0	0	0	0	MAP3	MAP2	MAP1	MAP0

MAP3=1: プレーン 3 への書き込み許可

MAP2=1: プレーン 2 への書き込み許可

MAP1=1: プレーン 1 への書き込み許可

MAP0=1: プレーン 0 への書き込み許可

[文字マップ選択レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
03H	0	0	CSA0	CSB0	CSA		CSB	

CSA0: 文字セット A のビット 0

CSB0: 文字セット B のビット 0

CSA : 文字セット A のビット 2 ~ 1

CSB : 文字セット B のビット 2 ~ 1

[メモリ・モード・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
04H	0	0	0	0	CHN4	CHN2	EXME	0

CHN4 = 0 : 4 プレーン連続指定

CHN2 = 0 : 2 プレーン連続指定

EXME = 1 : 拡張メモリ使用

(5) ディスプレイ・コントローラ

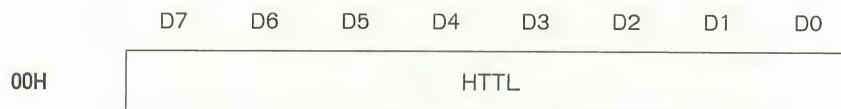
以下に、ディスプレイ・コントローラのレジスタを示します。

(モノクロ・モード：インデックス=3B4H, データ=3B5H)

(カラー・モード：インデックス=3D4H, データ=3D5H)

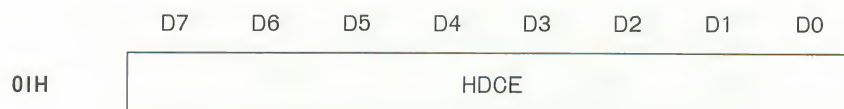
INDX	R/W	機 能
00H	R/W	水平総数レジスタ
01H	R/W	水平表示イネーブル終了レジスタ
02H	R/W	水平ブランキング開始レジスタ
03H	R/W	水平ブランキング終了レジスタ
04H	R/W	水平帰線開始レジスタ
05H	R/W	水平帰線終了レジスタ
06H	R/W	垂直総数レジスタ
07H	R/W	コントローラ・オーバーフロー・レジスタ
08H	R/W	プリセット行走査レジスタ
09H	R/W	最大走査線レジスタ
0AH	R/W	カーソル開始レジスタ
0BH	R/W	カーソル終了レジスタ
0CH	R/W	開始アドレス上位レジスタ
0DH	R/W	開始アドレス下位レジスタ
0EH	R/W	カーソル位置上位レジスタ
0FH	R/W	カーソル位置下位レジスタ
10H	R/W	垂直帰線開始レジスタ
11H	R/W	垂直帰線終了レジスタ・マップ
12H	R/W	垂直表示イネーブル終了レジスタ
13H	R/W	オフセット・レジスタ
14H	R/W	下線位置レジスタ
15H	R/W	垂直ブランキング開始レジスタ
16H	R/W	垂直ブランキング終了レジスタ
17H	R/W	CRTC モード制御レジスタ
18H	R/W	ライン比較レジスタ

[水平総数レジスタ]



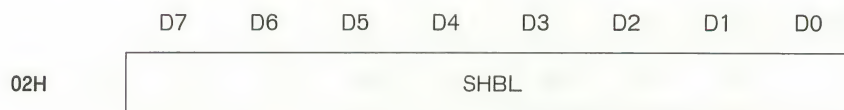
HTTL : 水平走査期間内の総文字数

[水平表示イネーブル終了レジスタ]



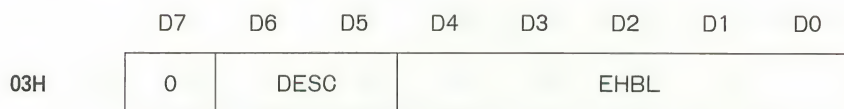
HDCE : 水平走査期間内の表示文字数-1

[水平ブランキング開始レジスタ]



SHBL : 水平ブランキング開始位置

[水平ブランキング終了レジスタ]



DESC : 表示イネーブル・スキュー

EHLB : 水平ブランキング終了位置

[水平帰線開始レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
04H	SHRT							

SHRT：水平帰線開始文字位置

[水平帰線終了レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
05H	EHB5	HRSK		EHBL				

EHB5：水平ブランキング終了位置の第5ビット

HRSK：水平帰線スキュー

EHBL：水平ブランキング終了位置

[垂直総数レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
06H	VTTL							

VTTL：(垂直走査線数-2)の下位8ビット

[コントローラ・オーバーフロー・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
07H	VRS9	VDE9	VTT9	LCM8	SVB8	VRS8	VDE8	VTT8

VRS9：垂直帰線開始位置の第9ビット

VDE9：垂直表示イネーブル終了位置の第9ビット

VTT9：(垂直走査線数-2)の第9ビット

LCM8：ライン比較の第8ビット

SVB8：垂直ブランキング開始位置の第8ビット

VRS8：垂直帰線開始位置の第8ビット

VDE8：垂直表示イネーブル終了位置の第8ビット

VTT8：(垂直走査線数-2)の第8ビット

[プリセット行走査レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
08H	0	BPCT		SRSC				

BPCT : 水平スクロールのバイト単位指定

SRSC : 表示開始ライン位置

[最大走査線レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
09H	LCNV	LCM9	SVB9	MXSL				

LCNV : 200 ライン 2 重表示

LCM9 : ライン比較の第 9 ビット

SVB9 : 垂直ブランキング開始位置の第 9 ビット

MXSL : 文字の縦ライン数-1

[カーソル開始レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
0AH	0	0	C0FF	CURS				

C0FF=1 : カーソル OFF

CURS : カーソル表示ライン位置-1

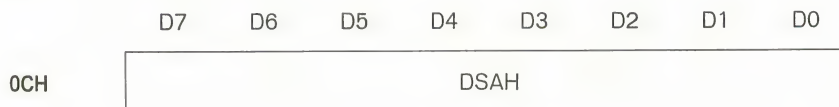
[カーソル終了レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
0BH	0	CSCT		CURE				

CSCT : カーソルのスキュー

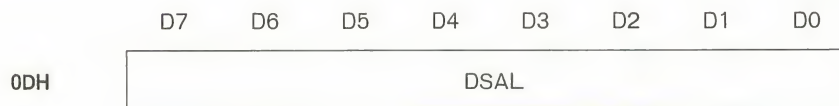
CURE : カーソル終了ライン位置-1

[開始アドレス上位レジスタ]



DSAH : 表示開始アドレスの上位 8 ビット

[開始アドレス下位レジスタ]



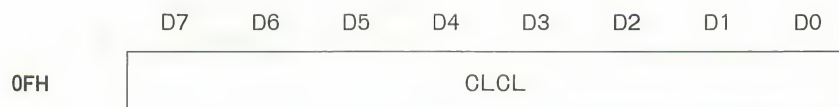
DSAL : 表示開始アドレスの下位 8 ビット

[カーソル位置上位レジスタ]



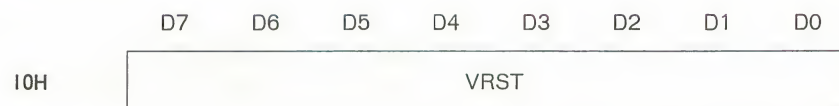
CLCH : カーソル位置の上位 8 ビット

[カーソル位置下位レジスタ]



CLCL : カーソル位置の下位 8 ビット

[垂直帰線開始レジスタ]



VRST : 垂直開始レジスタの下位 8 ビット

[垂直帰線終了レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
11H	PRTR	S5RC	EVIT	CVIT	VRTE			

PRTR=1: レジスタ 0～7 の書き込み禁止

S5RC : 水平周波数の選択

EVIT =0: 垂直帰線割り込みイネーブル

CVIT =0: 垂直帰線割り込みクリアー

VRTE : 垂直帰線終了ライン数

[垂直表示イネーブル終了レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
12H	VDEE							

VDEE: 垂直終了ラインの下位 8 ビット

[オフセット・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
13H	LOFS							

LOFS: 1 ライン当たりの幅

[下線位置レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
14H	0	AD32	CLK4	ULL0				

AD32=1: 32 ビットアドレス指定

CLK4=1: クロック 4 分周指定

ULL0 : アンダーラインの表示ライン数

[垂直ブランキング開始レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
15H	SVBL							

SVBL：垂直ブランキング開始ラインの下位 8 ビット

[垂直ブランキング終了レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
16H	EVBL							

EVBL：垂直ブランキング終了ラインの下位 8 ビット

[CRTC モード・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
17H	HDRS	ADMD	ADWP	0	CLK2	HRS2	SRSC	DPMD

HDRS = 1：垂直水平帰線許可

ADMD = 0：16 ビット・アドレス・モード

ADWP：折り返しアドレスの指定

CLK2 = 1：クロック 2 分周

HRS2 = 1：垂直幅 2 倍

SRSC = 1：垂直ライン・アドレス指定

DPMD = 1：インタレース表示

[ライン比較レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
18H	LCMP							

LCMP：画面分割ライン位置の下位 8 ビット

(6) グラフィック・コントローラ

以下に、グラフィック・コントローラのレジスタを示します。

(インデックス=3CEH, データ=3CFH)

INDX	R/W	機 能
00H	R/W	セット/リセット・レジスタ
01H	R/W	イネーブル・セット/リセット・レジスタ
02H	R/W	色比較レジスタ
03H	R/W	データ回転レジスタ
04H	R/W	リード・プレーン選択レジスタ
05H	R/W	グラフィック・モード・レジスタ
06H	R/W	多目的レジスタ
07H	R/W	色比較除外レジスタ
08H	R/W	ビット・マスク・レジスタ

[セット/リセット・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
00H	0	0	0	0	SRP3	SRP2	SRP1	SRP0

SRP3=1: プレーン 3 ビット・セット

SRP2=1: プレーン 2 ビット・セット

SRP1=1: プレーン 1 ビット・セット

SRP0=1: プレーン 0 ビット・セット

[イネーブル・セット/リセット・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
01H	0	0	0	0	ENP3	ENP2	ENP1	ENP0

ENP3=1: プレーン 3 書き込み許可

ENP2=1: プレーン 2 書き込み許可

ENP1=1: プレーン 1 書き込み許可

ENP0=1: プレーン 0 書き込み許可

[色比較レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
02H	0	0	0	0	GCP3	GCP2	GCP1	GCP0

GCP3=1: プレーン 3 と比較するビット

GCP2=1: プレーン 2 と比較するビット

GCP1=1: プレーン 1 と比較するビット

GCP0=1: プレーン 0 と比較するビット

[データ回転レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
03H	0	0	0	FNSL			RCNT	

FNSL=論理演算処理

00: 演算処理しない

01: AND(論理和)

10: OR(論理積)

11: XOR(排他的論理和)

RCNT=右回転するカウント数

[リード・プレーン選択レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
04H	0	0	0	0	0	0	PLSL	

PLSL=データをリードするプレーン番号

[グラフィック・モード・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
05H	0	M256	SFRM	OEMD	RDMD	0	WRMD	

M256 = 1 : 256 色モード

SFRM = 1 : シリアル・モード (CGA 互換)

OEMD = 1 : 奇数/偶数アドレス・モード (CGA 互換)

RDMD = 読み込みモード

0 : リード・マップ選択レジスタが有効

1 : 色比較レジスタが有効

WRMD = 書き込みモード

00 : 通常モード

01 : ラッチ・モード

10 : 塗りつぶしモード

11 : 文字描画モード

[多目的レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
06H	0	0	0	0	VADR		OEMD	GPMD

VADR = VRAM アドレス

00 : A0000H (128KB)

01 : A0000H (64KB)

10 : B0000H (32KB)

11 : B8000H (32KB)

OEMD = 1 : 奇数/偶数アドレス・モード (CGA 互換)

GPMD = 0 : 文字モード

1 : グラフィック・モード

[色比較除外レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
07H	0	0	0	0	CDP3	CDP2	CDP1	CDP0

CDP3 = 1 : プレーン 3 と比較する

CDP2 = 1 : プレーン 2 と比較する

CDP1 = 1 : プレーン 1 と比較する

CDP0 = 1 : プレーン 0 と比較する

[ビット・マスク・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
08H	MKB7	MKB6	MKB5	MKB4	MKB3	MKB2	MKB1	MKB0

MKB7=1: ビット 7 を操作許可
 MKB6=1: ビット 6 を操作許可
 MKB5=1: ビット 5 を操作許可
 MKB4=1: ビット 4 を操作許可
 MKB3=1: ビット 3 を操作許可
 MKB2=1: ビット 2 を操作許可
 MKB1=1: ビット 1 を操作許可
 MKB0=1: ビット 0 を操作許可

① 書き込みモード 0

書き込みモード 0 では、4 プレーンに対して事前に色の指定を行っておけば、以後は単一動作でドットやラインの描画を行うことができます。

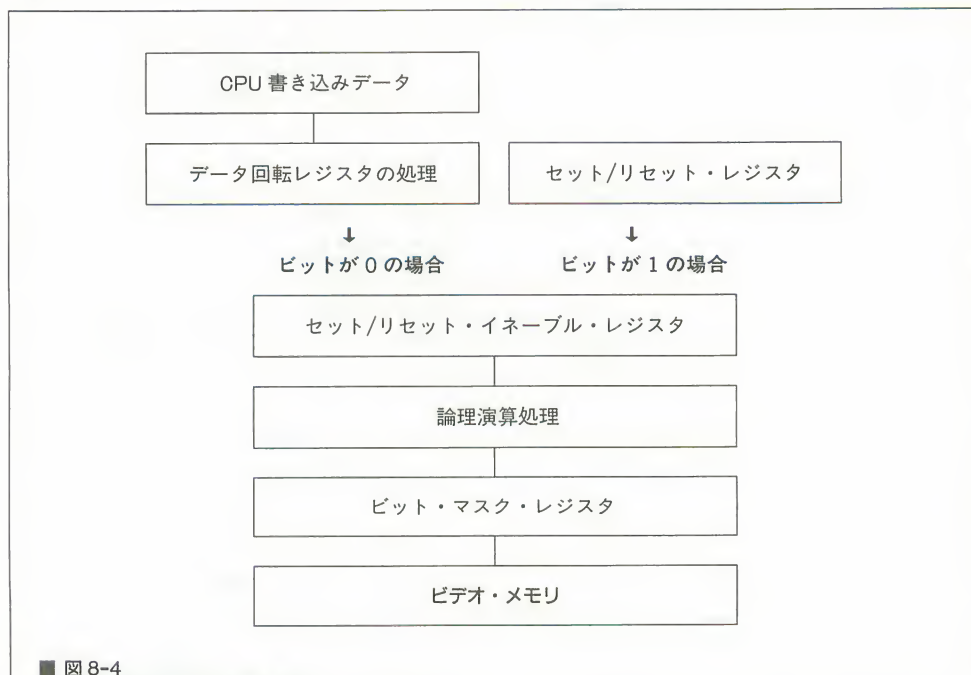


図 8-4

書き込みモード 0 の基本動作

List 8-3

書き込みモード 0 でのドット描画

```

;
; ES: DI = 描画アドレス  CL = ドット位置  BH = 色番号
;
DRAW_DOT:
    MOV     DX, 3CEH

    MOV     AX, 0005H           ; 書き込みモード 0
    OUT     (DX), AX

    MOV     AH, BH             ; 色番号設定
    MOV     AL, 00H
    OUT     (DX), AX

    MOV     AX, 0F01H          ; 4プレーン描画指定
    OUT     (DX), AL

    MOV     AX, 0003H          ; ファンクションモード
    OUT     (DX), AX

    MOV     AH, CL             ; ドット・データ設定
    MOV     AL, 08H
    OUT     (DX), AX

    OR      ES:BYTE PTR [DI], AL ; ダミー書き込み

    RET

```

② 書き込みモード 1

書き込みモード 1 では、読み書きに際して 4 プレーン全部が移動します。この機能で画面のスクロールや転送が高速に行えます。CPU のライトデータは無視され、転送はバイト単位しかできません。データは内部ラッチに保持されているものが使用されるので、少なくとも最初に 1 回は読み込み動作を行わないと、内部ラッチの内容が不定となってしまいます。

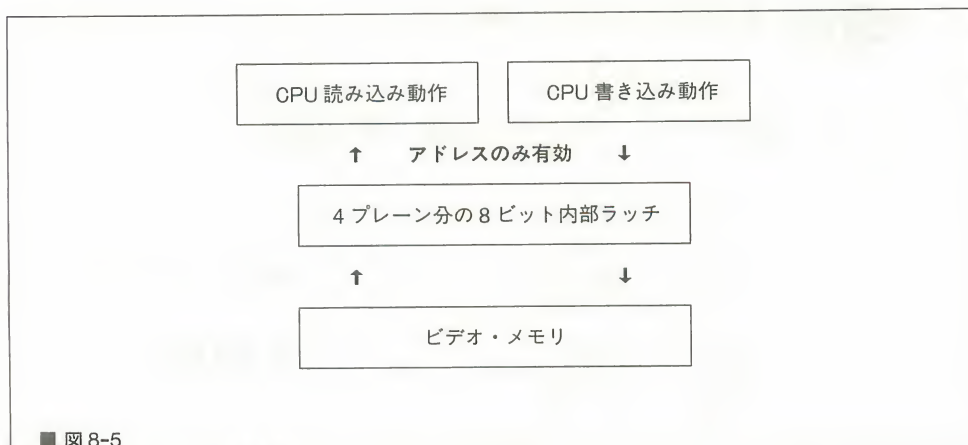


図 8-5

書き込みモード 1 の基本動作

List 8-4

書き込みモード 1 でのブロック転送

```

;
; DS : SI = 転送元  ES : DI = 転送先
; BX = 横バイト数  CX = 縦ライン数
;
BLOCK_MOVE:
    MOV     DX, 3CEH

    MOV     AX, 0105H                ; 書き込みモード 1
    OUT     (DX), AX

    CLD                                ; データ転送

BLOCK_MOVE_L1:
    PUSH    CX
    PUSH    SI
    PUSH    DI

    MOV     CX, BX                    ; 1 ラスタ転送
    REP     MOVSB

    POP     DI
    POP     SI
    POP     CX
    ADD     SI, 80
    ADD     DI, 80
    LOOP    BLOCK_MOVE_L1

    RET

```

③ 書き込みモード 2

書き込みモード 2 では、CPU の書き込みデータはセット/リセット・レジスタの値と同様に扱われ、指定のプレーンへそのビットが拡張されて描画が行われます。この機能では面の塗りつぶしなどが行えます。

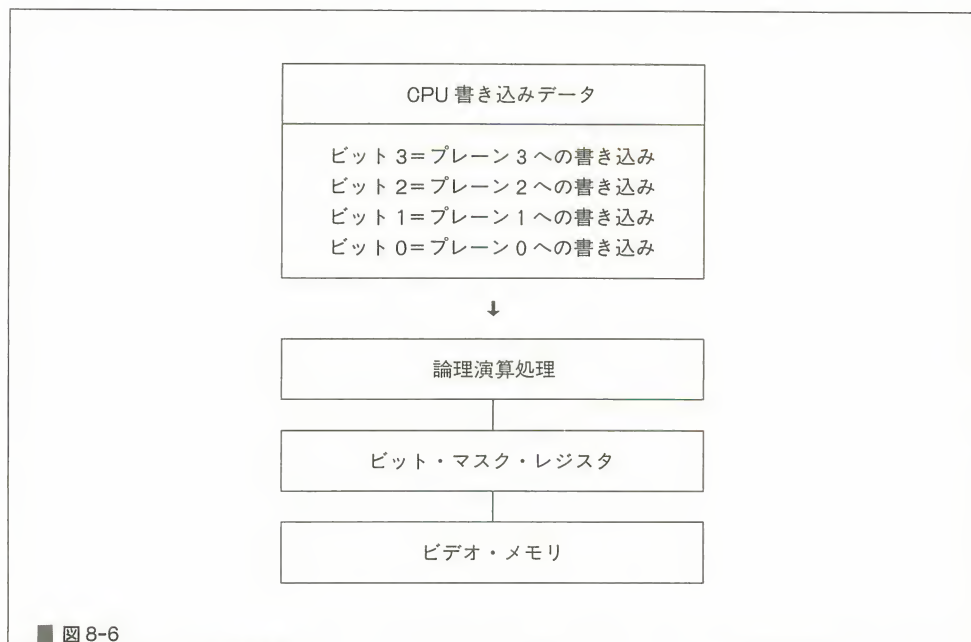


図 8-6

書き込みモード 2 の基本動作

List 8-5

書き込みモード 2 でのボックス描画

```

;
; ES : DI = BOX の左上座標  AL = 色番号
; BX = 横バイト数  CX = 縦ライン数
;
DRAW_BOX:
        PUSH    AX

        MOV     DX, 3CEH

        MOV     AX, 0105H                ; 書き込みモード 2
        OUT     (DX), AX

        MOV     AX, 0003H                ; ファンクションモード
        OUT     (DX), AX
  
```

	MOV	AX, 0007H	; 色比較レジスタ
	OUT	(DX), AX	
	POP	AX	
	CLD		; データ転送
DRAW_BOX_L1:	PUSH	CX	
	PUSH	DI	
	MOV	CX, BX	; 1ラスタ描画
	REP	STOSB	
	POP	DI	
	POP	CX	
	ADD	DI, 80	
	LOOP	DRAW_BOX_L1	
	RET		

④ 書き込みモード 3

書き込みモード 3 では、CPU データとビット・マスク・レジスタの値が AND され、ビットが 1 の部分のみがセット/リセット・レジスタの値に設定されます。この機能では文字の描画等を高速に行うことができます。

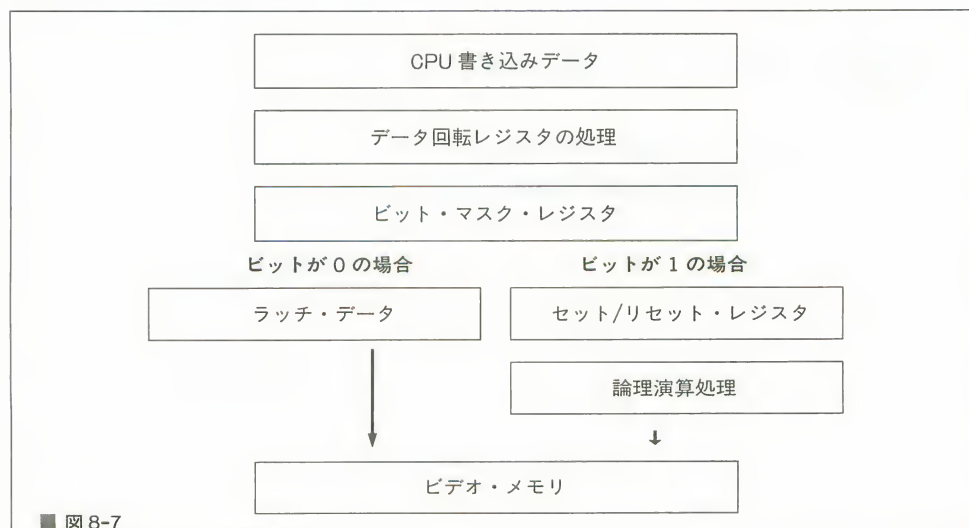


図 8-7

書き込みモード 3 の基本動作

List 8-6

書き込みモード 3 での文字描画

```

;
; DS: SI = 文字フォント   ES: DI = 描画アドレス   BH = 色番号
;
DRAW_FONT:
        MOV     DX, 3CEH

        MOV     AX, 0005H           ; 書き込みモード 3
        OUT     (DX), AX

        MOV     AH, BH             ; 色番号設定
        MOV     AL, 00H
        OUT     (DX), AX

        MOV     AX, 0003H           ; ファンクションモード
        OUT     (DX), AX

        MOV     AX, 0FF08H         ; マスク設定
        OUT     (DX), AX

        CLD
        MOV     CX, 8
DRAW_FONT_L1:
        MOVSB
        LOOP    DRAW_FONT_L1

        RET

```

(7) 属性コントローラ

属性コントローラのレジスタ設定は、インデックスおよびデータの順に同一のポート・アドレス (3C0H) へ書き込みます。

属性コントローラは、インデックス指定とデータ指定を示すラッチ (トグル・スイッチ) を内部的にもっており、これが交互に切り替わっています。このラッチは汎用レジスタのステータス・レジスタ 1 を読み込むことでリセットされます。

属性レジスタへの書き込み			
CLI			; 割り込み禁止
MOV	DX, 3DAH		; ラッチ・リセット
IN	AL, (DX)		
MOV	DX, 3C0H		; インデックス設定
MOV	AL, インデックス		
OUT	(DX), AL		
MOV	AL, データ		; データ書き込み
OUT	(DX), AL		
STI			; 割り込み許可

属性レジスタからの読み込み			
CLI			; 割り込み禁止
MOV	DX, 3DAH		; ラッチ・リセット
IN	AL, (DX)		
MOV	DX, 3C0H		; インデックス設定
MOV	AL, インデックス		
OUT	(DX), AL		
MOV	DX, 3C1H		; データ読み込み
IN	AL, (DX)		
STI			; 割り込み許可

以下に、属性コントローラのレジスタを示します。

(インデックス=3C0H, 書き込み=3C0H, 読み込み=3C1H)

INDX	R/W	機 能
00H	R/W	パレット・レジスタ
↓		
0FH		
10H	R/W	属性モード制御レジスタ
11H	R/W	オーバースキャン色レジスタ
12H	R/W	色プレーン・イネーブル・レジスタ
13H	R/W	水平ドット・パニング・レジスタ
14H	R/W	色選択レジスタ

[パレット・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
00H ↓ 0FH	0	0	PLTC					

PLTC=パレット番号

[属性モード制御レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
10H	ECSR	M256	HSCR	0	SCAT	ELNC	SMCE	GPMD

ECSR = 0 : パレット・レジスタ有効

1 : 色選択レジスタ有効

M256 = 1 : 256 色モード

HSCR = 1 : 水平ドット・パニング・レジスタ有効

SCAT = 文字属性のビット 7 の指定

0 : 背景輝度

1 : ブリンク

ELNC = 1 : 線グラフィック文字コード許可

SMCE = 0 : カラー・モード

1 : モノクロ・モード

GPMD = 0 : 文字モード

1 : グラフィック・モード

[オーバースキャン色レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
11H	0	0	OSPL					

OSPL=オーバースキャン・パレット

[色プレーン・イネーブル・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
12H	0	0	SMUX	ECP3	ECP2	ECP1	ECP0	

SMUX=ステータス・レジスタ 1 の診断用ビットへの値

ECP3=1: プレーン 3 表示許可

ECP2=1: プレーン 2 表示許可

ECP1=1: プレーン 1 表示許可

ECP0=1: プレーン 0 表示許可

[水平ドット・パニング・レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
13H	0	0	0	0	HSDC			

HSDC=水平ドット・スクロール数

[色選択レジスタ]

	D7	D6	D5	D4	D3	D2	D1	D0
14H	0	0	0	0	RDSL			

RDSL=RAMDAC レジスタ選択上位ビット

(8) ビデオ DAC

ビデオ DAC では、読み書きにインデックスを指定するポート・アドレスが異なります。一度インデックスを指定すると、データ・レジスタは読み書きのたびに自動的にインデックスが増加していくため、RGB アナログ・データを設定する場合などの連続したデータの設定が可能です。

ステータス・レジスタはこれらの設定に関係なく、いつでも読み出し可能で、読み書きの動作にも影響は与えません。

以下に、ビデオ DAC レジスタ I/O ポート・アドレスを示します。

読み込み ポート	書き込み ポート	機 能
3C8H	3C8H	書き込み用アドレス・レジスタ
----	3C7H	読み込み用アドレス・レジスタ
3C7H	----	ステータス・レジスタ
3C9H	3C9H	データ・レジスタ
3C6H	3C6H	ドット・マスク・レジスタ

[書き込み用アドレス・レジスタ]

D7	D6	D5	D4	D3	D2	D1	D0
WRRN							

WRRN=書き込み RAMDAC レジスタ番号

[読み込み用アドレス・レジスタ]

D7	D6	D5	D4	D3	D2	D1	D0
RDRN							

RDRN=書き込み RAMDAC レジスタ番号

[ステータス・レジスタ]

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	WRST	

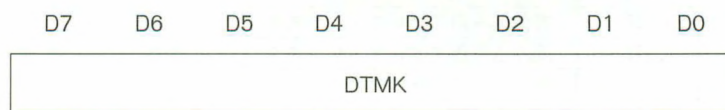
WRST=0:書き込み中

[データ・レジスタ]

D7	D6	D5	D4	D3	D2	D1	D0
0	0	CDLV					

CDLV=カラー・データ階調

[ドット・マスク・レジスタ]



DTMK = ドット・マスク (つねに FFH)

8.8 シリアル・ポート (NS16450)

シリアル・ポート・コントローラは、IBM-PC では INS8250A 相当品が使用されていましたが、現在では大半がそのフルコンパチブルの NS16450 を使用しています。このコントローラはプログラム可能な 2 つの非同期通信をサポートしており、主として RS-232C 制御に使用されています。

IBM-PC では 2 チャンネル (最大 4 チャンネル) が使用可能となっていますが、チャンネル 3、4 をどの I/O アドレスに設定するかは厳密には規定されていないので、BIOS のワーク (0040 : 0000H からの 4 ワード) を参照して、ベース・ポート・アドレスを取得すべきです。

以下に、I/O ポート・アドレスを示します。

● COM1 (割り込み先は IRQ4 : INTOCH)

ポート	R/W	機 能
3F8H	R/W	バッファ・レジスタ
	R/W	デバイザラッチ・レジスタ (下位バイト)
3F9H	W	通信割り込み制御レジスタ
	R/W	デバイザラッチ・レジスタ (上位バイト)
3FAH	R	通信割り込み表示レジスタ
3FBH	R/W	回線制御レジスタ
3FCH	R/W	モデム制御レジスタ
3FDH	R/W	回線ステータス・レジスタ
3FEH	R/W	モデム・ステータス・レジスタ

● COM2 (割り込み先は IRQ3 : INTOBH)

ポート	R/W	機 能
2F8H	R/W	バッファ・レジスタ
	R/W	デバイザラッチ・レジスタ (下位バイト)
2F9H	W	通信割り込み制御レジスタ
	R/W	デバイザラッチ・レジスタ (上位バイト)
2FAH	R	通信割り込み表示レジスタ
2FBH	R/W	回線制御レジスタ
2FCH	R/W	モデム制御レジスタ
2FDH	R/W	回線ステータス・レジスタ
2FEH	R/W	モデム・ステータス・レジスタ

(1) デバイザラッチ・レジスタ(リード/ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
3F8H 2F8H	BRCL							
3F9H 2F9H	BRCH							

BRCL=通信速度用内部クロック(下位バイト)

BRCH=通信速度用内部クロック(上位バイト)

通信速度	上位:下位	通信速度	上位:下位
50	09:00	600	00:C0
75	06:00	1200	00:60
110	04:17	2400	00:30
150	03:00	4800	00:18
300	01:80	9600	00:0C

(2) 通信割り込み制御レジスタ(ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
3F9H 2F9H	0	0	0	0	EDSI	ELSI	ETBE	ERBF

EDSI = 1: モデム・ステータス割り込み許可

ELSI = 1: 回線ステータス割り込み許可

ETBE = 1: 送信バッファ割り込み許可

ERBF = 1: データ受信割り込み許可

(3) 通信割り込み表示レジスタ(リード)

	D7	D6	D5	D4	D3	D2	D1	D0
3FAH 3FAH	0	0	0	0	0	IRMB	IRSF	

IRMB=割り込み識別ビット

00: モデム・ステータス割り込み

01: 送信バッファ割り込み

10: データ受信割り込み

11: 回線ステータス割り込み

IRSF=割り込み状況

1: なし

0: あり

(4) 回線制御レジスタ(リード/ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
3FBH 2FBH	DLAB	BRK	SPR	EPS	PEN	STB	DLEN	

DLAB=デバイザラッチ・アクセス

1:使用する

0:受信送信バッファ・レジスタ, 割り込み制御レジスタを使用

BRK =1:ブレーク信号送出

SPR =1:パリティ・ビットを固定する

EPS =パリティの種類

1:偶数パリティ

0:奇数パリティ

PEN =パリティの存在

STB =ストップビット長の選択

1:2ビット

0:1ビット

DLEN=データ長

0:5ビット, 1:6ビット, 2:7ビット, 3:8ビット

(5) モデム制御レジスタ(リード/ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
3FCH 2FCH	0	0	0	LOOP	OUT2	OUT1	RTS	DTR

LOOP=1:ローカル・ループバック・テスト

OUT2=ユーザ指定補助割り込み

1:割り込み許可

0:割り込み禁止

OUT1=ユーザ指定補助出力信号

RTS =1:RTS 信号制御

DTS =1:DTR 信号制御

(6) 回線ステータス・レジスタ(リード/ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
3FDH 2FDH	0	TEMT	THRE	BI	FE	PE	OE	RDR

TEMT=0: データ通信中

THRE=1: 送信バッファが空

BI =1: ブレーク信号受信

FE =1: フレーミング・エラー

PE =1: パリティ・エラー

OE =1: オーバーラン・エラー

RDR =1: 受信データ・イネーブル

(7) モデム・ステータス・レジスタ(リード/ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
3FEH 2FEH	DCD	CI	DSR	CTS	DDCD	TECI	DDSR	DCTS

DCD =DCD 信号

CI =CI 信号

DSR =DSR 信号

CTS =CTS 信号

DDCD=1: DCD 信号に変化あり

TECI =1: CI 信号に変化あり

DDSR=DSR 信号に変化あり

DCTS=CTS 信号に変化あり

8.9 パラレル・ポート (i8255A)

パラレル・ポート・コントローラは i8255A 相当品が使用されています。

これは主にプリンタ・ポートとして使用されていますが、最近ではストリーマなどでも使用されたりしています。

IBM-PC では 3 チャンネルが使用可能となっていますが、これらのパラレル・ポートをどの I/O アドレスに設定するかは厳密には規定されていないので、BIOS のワーク (0040:0008H からの 3 ワード) を参照して、ベース・ポート・アドレスを取得すべきです。

データ・レジスタは、設定により双方向へ読み書きできますが、これは単にバッファ内容が読み込めるだけであって、双方向転送ができるわけではないので注意が必要です。

以下に、I/O ポート・アドレス (一般的な場合) を示します。

● LPT1 (割り込みは IRQ7:INT0FH)

ポート	R/W	機 能
3BCH	R/W	データ・レジスタ
3BDH	R	ステータス・レジスタ
3BEH	R/W	コントロール・レジスタ

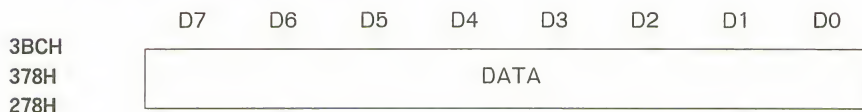
● LPT2 (割り込みは IRQ7:INT0FH)

ポート	R/W	機 能
378H	R/W	データ・レジスタ
379H	R	ステータス・レジスタ
37AH	R/W	コントロール・レジスタ

● LPT3 (割り込みは IRQ5:INT0DH)

ポート	R/W	機 能
278H	R/W	データ・レジスタ
279H	R	ステータス・レジスタ
27AH	R/W	コントロール・レジスタ

(1) データ・レジスタ (リード/ライト)



DATA = 出力データ

(2) ステータス・レジスタ(リード)

	D7	D6	D5	D4	D3	D2	D1	D0
3BDH 379H 279H	BUSY	ACK	PEMP	SLCT	ERR	0	0	0

BUSY = 0 : プリンタが印字中(負論理)

ACK = 0 : アクノリッジ信号を受けた(負論理)

PEMP = 1 : 用紙切れ

SLCT = 1 : プリンタがセレクトされた

ERR = 0 : プリンタ・エラーが発生(負論理)

(3) コントロール・レジスタ(リード/ライト)

	D7	D6	D5	D4	D3	D2	D1	D0
3BEH 37AH 27AH	0	0	DIO	IRQE	SLIN	IPRT	ATFD	STRB

DIO = 0 : データ・レジスタをライトのみで使用

1 : データ・レジスタをリード/ライトで使用

IRQE = 1 : ACK による割り込み許可

SLIN = 1 : プリンタ・セレクト出力

IPRT = 0 : プリンタ・リセット信号

ATFD = 1 : 1 行出力後の自動改行

STRB = 1 : ストロブ信号

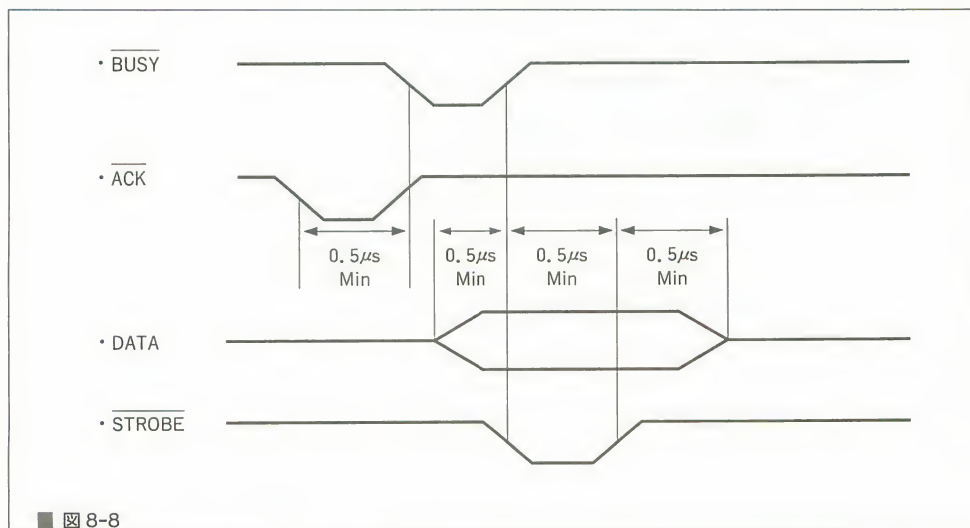


図 8-8

プリンタ出力のタイミング・シーケンス

List 8-7

簡単なプリンタ出力

```

;
; 文字列出力      DS:SI = 文字列(終端は0)
;
PUT_STR:
        CLD
PUT_STR_L1:
        LODSB
        OR      AL, AL
        JZ      PUT_STR_E
        CALL    PUT_CHR
        JMP     PUT_STR_L1

PUT_STR_E:
        RET

;
; 1文字出力      AL = 出力データ
;
PUT_CHR:
        PUSH    AL

        MOV     DX, 3BDH
PUT_CHR_L1:
        IN      AL, (DX)
        AND     AL, 10000000B
        JNZ     PUT_CHR_L1
        ; BUSY受信

        POP     AL
        ; データ出力

        MOV     DX, 3BCH
        OUT     (DX), AL
        ; ストローブ出力

        MOV     DX, 3BEH

        CALL    WAIT

        MOV     AL, 00000001B
        OUT     (DX), AL

        CALL    WAIT

        MOV     AL, 00000000B
        OUT     (DX), AL

        CALL    WAIT

```

RET

WAIT:

0.5 μ s 程度のウェイト

RET

8.10 ディスク・コントローラ

ディスク・コントローラはフロッピーディスク・コントローラが μ PD765となっていますが、ハードディスク・コントローラは厳密に規定がありません。

ディスク関係は、DOS のシステムからの操作を考えると、これを直接操作するメリットはほとんどありません。あるとすれば、ディスクにコピー・プロテクトをかけるときぐらいなので、ここでは詳細に触れず、I/O ポートの掲載にとどめておきます。

以下に、I/O ポート・アドレスを示します。

ポート	R/W	機 能
3F2H	W	FDD コントロール・レジスタ
3F4H	R	FDC ステータス・レジスタ
3F5H	R	FDC データ読み出し/結果ステータス・レジスタ
	W	FDC データ書き込み/コマンド・レジスタ
3F7H	R	FDD ステータス・レジスタ
	W	FDD コントロール・レジスタ

以下は、HDC の I/O ポート・アドレスです。

ポート	R/W	機 能
1F0H	R/W	データ・レジスタ (16 ビット)
1F1H	R	エラー・レジスタ
1F2H	R/W	セクタ数レジスタ
1F3H	R/W	セクタ番号レジスタ
1F4H	R/W	シリンダ番号レジスタ (下位)
1F5H	R/W	シリンダ番号レジスタ (上位)
1F6H	R/W	ドライブ・ヘッド番号レジスタ
1F7H	R	結果ステータス・レジスタ
	W	コマンド・レジスタ
3F6H	R	拡張ステータス・レジスタ
	W	コントロール・レジスタ
3F7H	R	ドライブ・アドレス・レジスタ



Appendix

Appendix A-1 文字コード表

DOS/V は日本語だけに限定されたシステムではありません。現在でもすでに中国語版・韓国語版が発表されており、これからより多くの国の言葉に対応していくことになるでしょう。

そこで本書では、IBM-PC が対応している多国語の文字コードを掲載しておきます。

(1) コード・ページ 932(日本語図形文字セット)

1	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	
	2 _A ↓ _B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		≡		0	@	P	'	p				ー	タ	ミ		
1	-1	⌈		!	1	A	Q	a	q				。	ア	チ	ム	
2	-2	⌋	↑	"	2	B	R	b	r				「	イ	ツ	メ	
3	-3	⌌		#	3	C	S	c	s				」	ウ	テ	モ	
4	-4	⌍	⌌	\$	4	D	T	d	t				、	エ	ト	ヤ	
5	-5	⌎	⌍	%	5	E	U	e	u				・	オ	ナ	ユ	
6	-6	⌏	⌎	&	6	F	V	f	v				ヲ	カ	ニ	ヨ	
7	-7	↓	⌏	'	7	G	W	g	w				ア	キ	ヌ	ラ	
8	-8			(8	H	X	h	x				イ	ク	ネ	リ	
9	-9	○	⌐)	9	I	Y	i	y				ウ	ケ	ノ	ル	
10	-A		⌑	*	:	J	Z	j	z				エ	コ	ハ	レ	
11	-B	☒	←	+	;	K	[k	{				オ	サ	ヒ	ロ	
12	-C		↑	,	<	L	¥	l					ヤ	シ	フ	ワ	
13	-D			-	=	M]	m	}				ユ	ス	ヘ	ン	
14	-E	■	→	.	>	N	^	n	—				ヨ	セ	ホ	"	
15	-F	☼	←	/	?	O	_	o					ッ	ソ	マ	・	

2バイト・コード文字の1バイト目

2バイト・コード文字の2バイト目

(2) コード・ページ 437 (IBM PC 図形文字セット)

1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	2 _{AB} ↓B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		▶		0	@	P	`	p	Ç	É	á	⋮	⌒	⌒	α	≡
1	-1	☺	◀	!	l	A	Q	a	q	ü	æ	í	⋮	⌒	⌒	β	±
2	-2	☹	↕	"	2	B	R	b	r	é	Æ	ó	⋮	⌒	⌒	Γ	≥
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú		⌒	⌒	π	≤
4	-4	♦	⚡	\$	4	D	T	d	t	ä	ö	ñ	⌒	⌒	⌒	Σ	f
5	-5	♣	§	%	5	E	U	e	u	à	ò	ñ	⌒	⌒	⌒	σ	J
6	-6	♠	—	&	6	F	V	f	v	å	û	ª	⌒	⌒	⌒	μ	÷
7	-7	•	↕	'	7	G	W	g	w	ç	ù	º	⌒	⌒	⌒	τ	≈
8	-8	■	↑	(8	H	X	h	x	ê	ÿ	¿	⌒	⌒	⌒	Φ	°
9	-9	○	↓)	9	I	Y	i	y	ë	Ö	⌒	⌒	⌒	⌒	⊙	•
10	-A	◉	→	*	:	J	Z	j	z	è	Ü	⌒	⌒	⌒	⌒	Ω	•
11	-B	♂	←	+	;	K	l	k	{	ï	¢	½	⌒	⌒	■	δ	√
12	-C	♀	⌒	,	<	L	\	l		î	£	¼	⌒	⌒	■	∞	ⁿ
13	-D	♪	↔	-	=	M	l	m	}	ì	¥	ì	⌒	⌒	■	φ	²
14	-E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	⌒	⌒	■	ε	■
15	-F	☼	▼	/	?	O	_	o	△	Å	f	»	⌒	⌒	■	η	

(3) コード・ページ 850(多国語図形文字セット)

1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	2 _A → ↓B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		▶		0	@	P	'	p	Ç	É	á	⋮	⌒	ð	Ó	-
1	-1	☺	◀	!	1	A	Q	a	q	ü	æ	í	⋮	⌒	Ð	β	±
2	-2	☺	↕	ˆ	2	B	R	b	r	é	Æ	ó	⋮	⌒	Ê	Ô	=
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú		⌒	È	Ò	¼
4	-4	♦	¶	\$	4	D	T	d	t	ä	ö	ñ		—	È	ö	€
5	-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	+	!	Ö	§
6	-6	♠	—	&	6	F	V	f	v	â	û	ª	Â	ã	Í	μ	÷
7	-7	•	↕	'	7	G	W	g	w	ç	ù	º	À	Ã	Î	þ	.
8	-8	■	↑	(8	H	X	h	x	ê	ÿ	ı	©	⌒	Ï	Ɔ	°
9	-9	○	↓)	9	I	Y	i	y	ë	Ö	®	⌒	⌒	⌒	Ú	..
10	-A	☉	→	*	:	J	Z	j	z	è	Ü	⌒	⌒	⌒	⌒	Û	•
11	-B	♂	←	+	;	K	l	k	{	ï	ø	½	⌒	⌒	■	Ü	¹
12	-C	♀	⌒	,	<	L	\	l		î	£	¼	⌒	⌒	■	ý	³
13	-D	♪	↔	-	=	M]	m	}	ì	Ø	ı	¢	=	ı	Ÿ	²
14	-E	♫	▲	.	>	N	^	n	~	Ä	x	«	¥	⌒	ı	-	■
15	-F	☀	▼	/	?	O	_	o	△	Å	f	»	⌒	⌒	■	,	

(4) コード・ページ 852(スラブ語図形文字セット)

1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	2 _A ↓ _B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		▶		0	@	P	`	p	Ç	É	á	⋮	Ł	đ	Ó	—
1	-1	☺	◀	!	l	A	Q	a	q	ü	Ł	í	⋮	Ł	Đ	β	"
2	-2	☺	↑↓	"	2	B	R	b	r	é	Í	ó	⋮	Ł	Đ	Ô	„
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú		Ł	Ě	Ň	ˇ
4	-4	♦	◀	\$	4	D	T	d	t	ä	ö	Ą	Ł	Ł	đ	ń	˘
5	-5	♣	§	%	5	E	U	e	u	û	Ł	ą	Ą	Ł	Ň	ň	§
6	-6	♠	—	&	6	F	V	f	v	ć	İ	ž	Ā	Ā	İ	Š	÷
7	-7	•	↑↓	'	7	G	W	g	w	ç	Š	ž	Ě	ā	İ	š	,
8	-8	■	↑	(8	H	X	h	x	ı	š	Ę	Ś	Ł	ě	Ř	°
9	-9	○	↓)	9	I	Y	i	y	ě	Ö	ę	Ł	Ł	Ł	Ú	¨
10	-A	◎	→	*	:	J	Z	j	z	Ō	Ü	Ł	Ł	Ł	Ł	ı	•
11	-B	♂	←	+	:	K	[k	{	ō	Ť	ž	Ł	Ł	■	Ů	ũ
12	-C	♀	Ł	.	<	L	\	l		î	ĩ	Č	Ł	Ł	■	ý	Ř
13	-D	♪	↔	-	=	M]	m	}	Ž	Ł	š	Ž	=	Ł	Ý	ř
14	-E	♪	▲	.	>	N	^	n	~	Ä	x	«	ž	Ł	Ł	Ł	■
15	-F	☼	▼	/	?	O	—	o	△	Č	č	»	Ł	Ł	■	'	

(5) コード・ページ 857(トルコ語図形文字セット)

1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	2A ↓B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		▶		0	@	P	`	p	Ç	É	á	▤	└	º	Ó	—
1	-1	☺	◀	!	l	A	Q	a	q	ü	æ	í	▥	┐	ª	β	±
2	-2	☺	↕	"	2	B	R	b	r	é	Æ	ó	▨	└	Ê	Ô	
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú		└	È	Ò	¼
4	-4	♦	¶	\$	4	D	T	d	t	ä	ö	ñ	└	—	È	ö	ƒ
5	-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	+		Õ	§
6	-6	♠	—	&	6	F	V	f	v	å	û	Ğ	Â	ã	í	μ	÷
7	-7	•	↕	'	7	G	W	g	w	ç	ù	ğ	À	Ã	Î		,
8	-8	■	↑	(8	H	X	h	x	ê	î	ı	©	└	ÿ	×	°
9	-9	○	↓)	9	I	Y	i	y	ë	Ö	@	└	└	└	Ú	..
10	-A	◉	→	*	:	J	Z	j	z	è	Ü	└		└	└	Û	•
11	-B	♂	←	+	;	K	l	k	{	ĩ	ø	½	└	└	■	Ù	¹
12	-C	♀	└	,	<	L	\	l		î	£	¼	└	└	■	ì	³
13	-D	♪	↔	-	=	M	l	m	}	l	Ø	i	¢	=		ÿ	²
14	-E	♪	▲	.	>	N	^	n	~	Ä	Ş	«	¥	└	ì	—	■
15	-F	☼	▼	/	?	O	—	o	△	Å	ş	»	└	◊	■	,	

(6) コード・ページ 860(ポルトガル語図形文字セット)

1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	2 ^A ↓B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		▶		0	@	P	`	p	Ç	É	á	⋮	⌒	⌒	α	≡
1	-1	☺	◀	!	l	A	Q	a	q	ü	À	í	⋮	⌒	⌒	β	±
2	-2	☹	↑↓	"	2	B	R	b	r	é	È	ó	⋮	⌒	⌒	Γ	≥
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú		⌒	⌒	π	≤
4	-4	♦	⚡	\$	4	D	T	d	t	ã	õ	ñ	⌒	⌒	⌒	Σ	f
5	-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	⌒	⌒	⌒	σ	J
6	-6	♠	—	&	6	F	V	f	v	Á	Ú	ª	⌒	⌒	⌒	μ	÷
7	-7	•	↑↓	'	7	G	W	g	w	ç	ù	º	⌒	⌒	⌒	τ	≈
8	-8	■	↑	(8	H	X	h	x	ê	î	¿	⌒	⌒	⌒	Φ	°
9	-9	○	↓)	9	I	Y	i	y	Ê	Ë	Ò	⌒	⌒	⌒	Θ	•
10	-A	◉	→	*	:	J	Z	j	z	è	Ü	⌒	⌒	⌒	⌒	Ω	•
11	-B	♂	←	+	;	K	[k	{	Í	ø	½	⌒	⌒	■	δ	√
12	-C	♀	⌒	,	<	L	\	l		Ô	£	¼	⌒	⌒	■	∞	n
13	-D	♪	↔	-	=	M]	m	}	ì	Ù	ì	⌒	⌒	■	φ	²
14	-E	🎵	▲	.	>	N	^	n	~	Ã	Pt	«	⌒	⌒	■	ε	■
15	-F	☼	▼	/	?	O	_	o	△	Â	Ó	»	⌒	⌒	■	∩	

(7) コード・ページ 861(アイスランド語図形文字セット)

1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	2 _A ↓ _B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		▶		0	@	P	`	p	Ç	É	á	⋮	└	ð	Ó	-
1	-1	☺	◀	!	l	A	Q	a	q	ü	æ	í	⋮	└	Ð	β	±
2	-2	☺	↑↓	"	2	B	R	b	r	é	Æ	ó	⋮	└	Ê	Ô	=
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú		└	È	Ò	¼
4	-4	♦	⚡	\$	4	D	T	d	t	ä	ö	ñ	└	—	È	ô	¢
5	-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	+	ı	Õ	§
6	-6	♠	—	&	6	F	V	f	v	å	û	ª	Â	ã	í	μ	÷
7	-7	•	↑↓	'	7	G	W	g	w	ç	ù	º	À	Ã	î	þ	,
8	-8	■	↑	(8	H	X	h	x	ê	ÿ	¿	©	└	ÿ	þ	°
9	-9	○	↓)	9	I	Y	i	y	ë	Ö	®	└	└	└	Ú	..
10	-A	◉	→	*	:	J	Z	j	z	è	Ü	¬		≡	└	Ù	•
11	-B	♂	←	+	;	K	[k	{	ï	ø	½	└	└	■	Ù	¹
12	-C	♀	└	,	<	L	\	l		î	£	¼	└	└	■	ý	³
13	-D	♪	↔	-	=	M]	m	}	ì	Ø	ı	¢	=	ı	Ý	²
14	-E	🎵	▲	.	>	N	^	n	~	Ä	x	«	¥	└	ı	—	■
15	-F	☀	▼	/	?	O	—	o	△	Å	f	»	└	¤	■	,	

(8) コード・ページ 863(カナダ・フランス語図形文字セット)

1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	$\begin{smallmatrix} 2\text{A} \\ \downarrow \text{B} \end{smallmatrix}$	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		▶		0	@	P	`	p	Ç	É	!	⋮	⌒	⌒	α	≡
1	-1	☺	◀	!	1	A	Q	a	q	ü	È	'	⋮	⌒	⌒	β	±
2	-2	☹	↕	"	2	B	R	b	r	é	Ê	ó	⋮	⌒	⌒	Γ	≥
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú			⌒	π	≤
4	-4	♦	↕	\$	4	D	T	d	t	Â	Ë	"		—	⌒	Σ	f
5	-5	♣	§	%	5	E	U	e	u	à	Ï	,	=	+	⌒	σ	J
6	-6	♠	—	&	6	F	V	f	v	¢	û	³		⌒	⌒	μ	÷
7	-7	•	↕	'	7	G	W	g	w	ç	ù	-	⌒	⌒	⌒	τ	≈
8	-8	■	↑	(8	H	X	h	x	ê	Ɔ	î	⌒	⌒	⌒	Φ	°
9	-9	○	↓)	9	I	Y	i	y	ë	Ô	⌒	⌒	⌒	⌒	Θ	•
10	-A	◉	→	*	:	J	Z	j	z	è	Ü	⌒	⌒	⌒	⌒	Ω	•
11	-B	♂	←	+	;	K	l	k	{	ĩ	¢	½	⌒	⌒	■	δ	✓
12	-C	♀	⌒	,	<	L	\	l		î	£	¼	⌒	⌒	■	∞	"
13	-D	♪	↔	-	=	M	l	m	}	=	Û	¾	⌒	⌒	■	φ	²
14	-E	♫	▲	.	>	N	^	n	~	À	Û	«	⌒	⌒	■	ε	■
15	-F	☼	▼	/	?	O	—	o	△	§	f	»	⌒	⌒	■	∩	

(9) コード・ページ 865(北欧語図形文字セット)

1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	2 ^A ↓B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		▶		0	@	P	`	p	Ç	É	á	⋮	⌒	⌒	α	≡
1	-1	☺	◀	!	l	A	Q	a	q	ü	æ	i	⋮	⌒	⌒	β	±
2	-2	☹	↕	"	2	B	R	b	r	é	Æ	ó	⋮	⌒	⌒	Γ	≥
3	-3	♥	!!	#	3	C	S	c	s	â	ô	ú		⌒	⌒	π	≤
4	-4	♦	☹	\$	4	D	T	d	t	ä	ö	ñ	⌒	⌒	⌒	Σ	f
5	-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	⌒	⌒	⌒	σ	J
6	-6	♠	—	&	6	F	V	f	v	å	û	ª	⌒	⌒	⌒	μ	÷
7	-7	•	↕	'	7	G	W	g	w	ç	ù	º	⌒	⌒	⌒	τ	≈
8	-8	■	↑	(8	H	X	h	x	ê	ÿ	¿	⌒	⌒	⌒	Φ	°
9	-9	○	↓)	9	I	Y	i	y	ë	Ö	⌒	⌒	⌒	⌒	Θ	•
10	-A	☉	→	*	:	J	Z	j	z	è	Ü	⌒	⌒	⌒	⌒	Ω	•
11	-B	♂	←	+	;	K	[k	{	ĩ	ø	½	⌒	⌒	⌒	δ	✓
12	-C	♀	⌒	,	<	L	\	l		î	£	¼	⌒	⌒	⌒	∞	ⁿ
13	-D	♪	↔	-	=	M]	m	}	ì	Ø	ì	⌒	⌒	⌒	φ	²
14	-E	🎵	▲	.	>	N	^	n	~	Ä	Pt	«	⌒	⌒	⌒	ε	■
15	-F	☼	▼	/	?	O	_	o	△	Å	f	⌒	⌒	⌒	⌒	∩	

(10) コード・ページ 869(ギリシャ語図形文字セット)

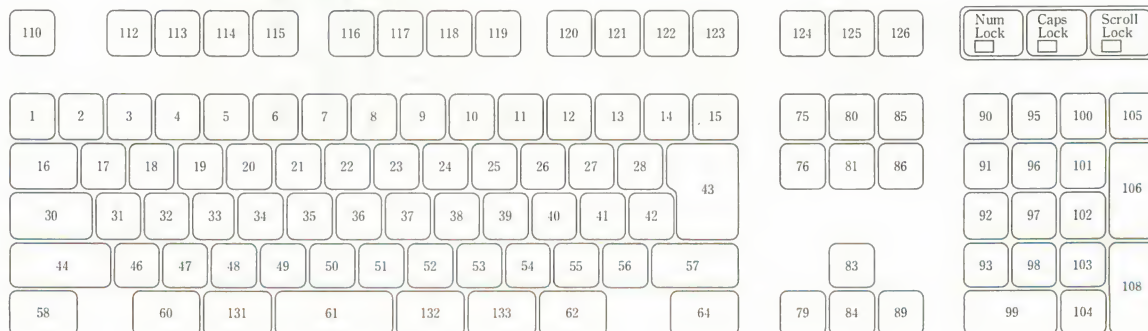
1		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	2A ↓B	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
0	-0		►		0	@	P	`	p		ı	ı	■	ı	ı	ı	-
1	-1	☺	◄	!	ı	A	Q	a	q		İ	İ	■	ı	ı	ı	ı
2	-2	☹	↕	ˆ	2	B	R	b	r		Ů	ó	■	ı	ı	ı	ı
3	-3	♥	!!	#	3	C	S	c	s			ú	ı	ı	ı	ı	ı
4	-4	♦	ı	\$	4	D	T	d	t			À	ı	-	Ψ	K	X
5	-5	♣	§	%	5	E	U	e	u		Υ	B	K	ı	Ω	ı	ı
6	-6	♠	—	&	6	F	V	f	v	À	ÿ	Γ	Λ	Π	α	μ	ψ
7	-7	•	↕	ˆ	7	G	W	g	w		©	Δ	Μ	Ρ	β	ν	ˆ
8	-8	■	↑	(8	H	X	h	x	•	Ω	E	N	ı	ı	ı	ı
9	-9	○	↓)	9	I	Y	i	y	ı	²	Z	ı	ı	ı	ı	ı
10	-A	⊙	→	*	:	J	Z	j	z	ı	³	H	ı	ı	ı	ı	ı
11	-B	♂	←	+	;	K	ı	k	(ı	á	½	ı	ı	ı	ı	ı
12	-C	♀	└	,	<	L	\	ı	ı	ı	£	θ	ı	ı	ı	ı	ı
13	-D	♪	↔	-	=	M	ı	m)	ı	É	I	ı	=	δ	ı	ı
14	-E	🎵	▲	.	>	N	^	n	~	-	ı	«	ı	ı	ı	ı	ı
15	-F	☀	▼	/	?	O	_	o	△	ı	ı	»	ı	ı	ı	ı	ı

Appendix A-2 IBM 5576-A01 キーボード

・キーボードの刻印



・キーボードのキー番号



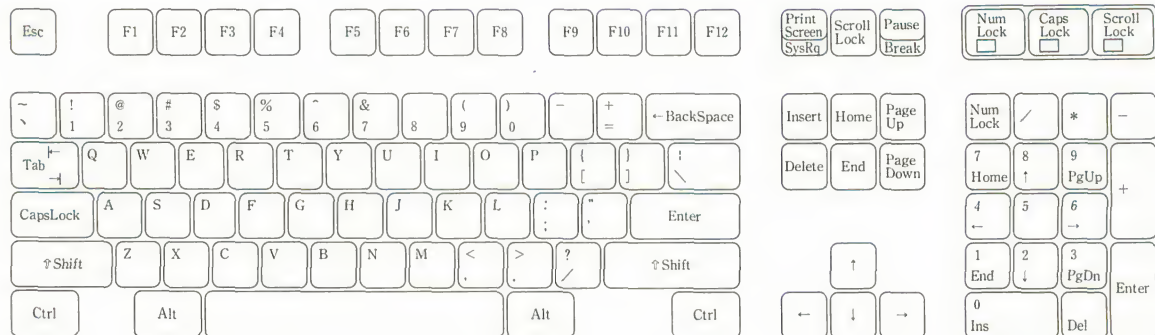
キー番号	刻 印	下 段	上 段 (Shift+)	Ctrl+	Alt+
1	半角/全角	***, AF/00	***, B0/00	***, B1/00	***, B2/00
2	1 ! め	1 02/31	! 02/21	***	78/00
3	2 " ふ	2 03/32	" 03/22	NUL 03/00	79/00
4	3 # あ	3 04/33	# 04/23	***	7A/00
5	4 \$ う	4 05/34	\$ 05/24	***	7B/00
6	5 % え	5 06/35	% 06/25	***	7C/00
7	6 & お	6 07/36	& 07/26	RS 07/1E	7D/00
8	7 ' や	7 08/37	' 08/27	***	7E/00
9	8 (ゆ	8 09/38	(09/28	***	7F/00
10	9) よ	9 0A/39) 0A/29	***	80/00
11	0 ~ わ	0 0B/30	***, 0B/00	***	81/00
12	= ほ	= 0C/2D	= 0C/3D	US 0C/1F	82/00
13	- - ヘ	- 6D/5E	- 00/7E	***	83/00
14	¥ - ー	¥ 7D/5C	7D/7C	FS 7D/1C	***
15	Backspace	BS 0E/08	BS 0E/08	DEL 0E/7F	***, 0E/00
16	Tab Backtab	HT 0F/09	0F/00	***, 94/00	***, A5/00
17	Q た	q 10/71	Q 10/51	DC1 10/11	10/00
18	W て	w 11/77	W 11/57	ETB 11/17	11/00
19	E い	e 12/65	E 12/45	ENQ 12/05	12/00
20	R す	r 13/72	R 13/52	DC2 13/12	13/00
21	T か	t 14/74	T 14/54	DC4 14/14	14/00
22	Y ん	y 15/79	Y 15/59	EM 15/19	15/00
23	U な	u 16/75	U 16/55	NAK 16/15	16/00
24	I に	i 17/69	I 17/49	HT 17/09	17/00
25	O ら	o 18/6F	O 18/4F	SI 18/0F	18/00
26	P せ	p 19/70	P 19/50	DLE 19/10	19/00
27	@ ' " c	@ 1A/40	' 1A/60	***	***, 1A/00
28	[{ " r	[1B/5B	{ 1B/7B	ESC 1B/1B	***, 1B/00
29	(キーなし)				
30	英数 CapsLock	***, B3/00	***	***, B4/00	***, B5/00
31	A ち	a 1E/61	A 1E/41	SOH 1E/01	1E/00
32	S と	s 1F/73	S 1F/53	DC3 1F/13	1F/00
33	D し	d 20/64	D 20/44	EOT 20/04	20/00
34	F は	f 21/66	F 21/46	ACK 21/06	21/00
35	G き	g 22/67	G 22/47	BEL 22/07	22/00
36	H く	h 23/68	H 23/48	BS 23/08	23/00
37	J ま	j 24/6A	J 24/4A	LF 24/0A	24/00
38	K の	k 25/6B	K 25/4B	VT 25/0B	25/00
39	L り	l 26/6C	L 26/4C	FF 26/0C	26/00
40	: + れ	: 27/3B	+ 27/2B	***	***, 27/00
41	: * け	: 28/3A	* 28/2A	***	***, 28/00
42] } む] 2B/5D	} 2B/7D	GS 2B/1D	***, 2B/00
43	Enter	CR 1C/0D	CR 1C/0D	LF 1C/0A	***, 1C/00
44	Shift	***	***	***	***
45	(キーなし)				
46	Z つ	z 2C/7A	Z 2C/5A	SUB 2C/1A	2C/00
47	X さ	x 2D/78	X 2D/58	CAN 2D/18	2D/00
48	C そ	c 2E/63	C 2E/43	ETX 2E/03	2E/00
49	V ひ	v 2F/76	V 2F/56	SYN 2F/16	2F/00
50	B こ	b 30/62	B 30/42	STX 30/02	30/00

51	N み	n	31/6E	N	31/4E	SO	31/0E	31/00
52	M も	m	32/6D	M	32/4D	CR	32/0D	32/00
53	, < ね ,	,	33/2C	<	33/3C		**/**	**/**, 33/00
54	. > る .	.	34/2E	>	34/3E		**/**	**/**, 34/00
55	/ ? め .	/	35/2F	?	35/3F		**/**	**/**, 35/00
56	\ _ ろ ;	¥	73/5C	—	73/5F	FS	73/1C	**/**
57	Shift		**/**		**/**		**/**	**/**
58	Ctrl		**/**		**/**		**/**	**/**
59	(キーなし)							
60	Alt		**/**		**/**		**/**	**/**
61	Space	SP	39/20	SP	39/20	SP	39/20	SP 39/20
62	Alt		**/**		**/**		**/**	**/**
63	(キーなし)							
64	Ctrl		**/**		**/**		**/**	**/**
65	(キーなし)							
66	(キーなし)							
67	(キーなし)							
68	(キーなし)							
69	(キーなし)							
70	(キーなし)							
71	(キーなし)							
72	(キーなし)							
73	(キーなし)							
74	(キーなし)							
75	Insert		52/00, 52/E0		52/00, 52/E0		**/**, 92/E0	**/**, A2/00
76	Delete		53/00, 53/E0		53/00, 53/E0		**/**, 93/E0	**/**, A3/00
77	(キーなし)							
78	(キーなし)							
79	←		4B/00, 4B/E0		4B/00, 4B/E0		73/00, 73/E0	**/**, 9B/00
80	Home		47/00, 47/E0		47/00, 47/E0		77/00, 77/E0	**/**, 97/00
81	End		4F/00, 4F/E0		4F/00, 4F/E0		75/00, 75/E0	**/**, 9F/00
82	(キーなし)							
83	↑		48/00, 48/E0		48/00, 48/E0		**/**, 8D/E0	**/**, 98/00
84	↓		50/00, 50/E0		50/00, 50/E0		**/**, 91/E0	**/**, A0/00
85	PageUp		49/00, 49/E0		49/00, 49/E0		84/00, 84/E0	**/**, 99/00
86	PageDown		51/00, 51/E0		51/00, 51/E0		76/00, 76/E0	**/**, A1/00
87	(キーなし)							
88	(キーなし)							
89	→		4D/00, 4D/E0		4D/00, 4D/E0		74/00, 74/E0	**/**, 9D/00
90	NumLock		**/**		**/**		**/**	**/**
91	Home 7		47/00	7	47/37		77/00	Code Input
92	← 4		4B/00	4	4B/34		73/00	Code Input
93	End 1		4F/00	1	4F/31		75/00	Code Input
94	(キーなし)							
95	/	/	35/2F, E0/2F	/	35/2F, E0/2F		**/**, 95/00	**/**, A4/00
96	↑ 8		48/00	8	48/38		**/**, 8D/00	Code Input
97	5		**/**, 4C/00	5	4C/35		**/**, 8F/00	Code Input
98	↓ 2		50/00	2	50/32		**/**, 91/00	Code Input
99	Ins 0		52/00	0	52/30		**/**, 92/00	Code Input
100	*	*	37/2A	*	37/2A		**/**, 96/00	**/**, 37/00
101	PgUp 9		49/00	9	49/39		84/00	Code Input

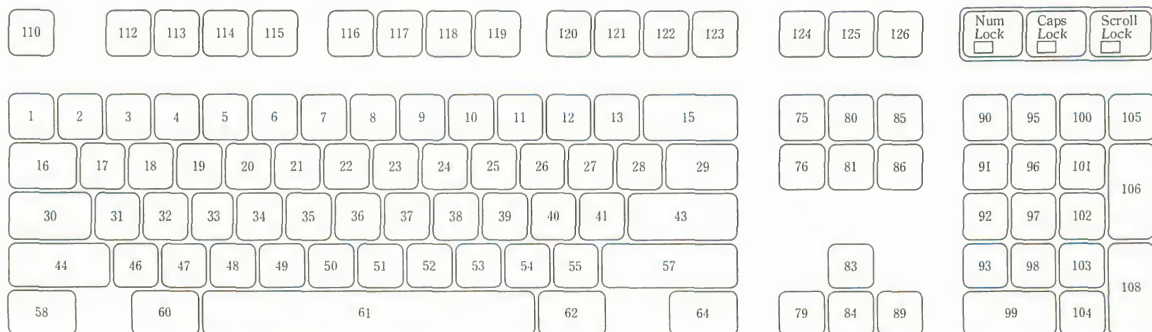
102	→	6	4D/00	6	4D/36	74/00	Code Input
103	PgDn	3	51/00	3	51/33	76/00	Code Input
104	Del	.	53/00	.	53/2E	**/**, 93/00	**/**
105		—	4A/2D	—	4A/2D	**/**, 8E/00	**/**, 4A/00
106		+	4E/2B	+	4E/2B	**/**, 90/00	**/**, 4E/00
107	(キーなし)						
108	Enter		CR 1C/0D, E0/0D	CR 1C/0D, E0/0D	LF 1C/0A, E0/0A	**/**, A6/00	
109	(キーなし)						
110	Esc		ESC 01/1B	ESC 01/1B	ESC 01/1B	**/**, 01/00	
111	(キーなし)						
112	F1		3B/00	54/00	5E/00	68/00	
113	F2		3C/00	55/00	5F/00	69/00	
114	F3		3D/00	56/00	60/00	6A/00	
115	F4		3E/00	57/00	61/00	6B/00	
116	F5		3F/00	58/00	62/00	6C/00	
117	F6		40/00	59/00	63/00	6D/00	
118	F7		41/00	5A/00	64/00	6E/00	
119	F8		42/00	5B/00	65/00	6F/00	
120	F9		43/00	5C/00	66/00	70/00	
121	F10		44/00	5D/00	67/00	71/00	
122	F11		**/**, 85/00	**/**, 87/00	**/**, 89/00	**/**, 8B/00	
123	F12		**/**, 86/00	**/**, 88/00	**/**, 8A/00	**/**, 8C/00	
124	PrintScreen		**/**	**/**	72/00	**/**	
125	ScrollLock		**/**	**/**	**/**	**/**	
126	Pause		**/**	**/**	00/00	**/**	
127	(キーなし)						
128	(キーなし)						
129	(キーなし)						
130	(キーなし)						
131	無変換		**/**, AB/00	**/**, AC/00	**/**, AD/00	**/**, AE/00	
132	変換 前候補		**/**, A7/00	**/**, A8/00	**/**, A9/00	**/**, AA/00	
133	ひらがな カタカナ		**/**, B6/00	**/**, B7/00	**/**, B8/00	**/**, B9/00	

Appendix A-3 IBM U.S.English キーボード

• キーボードの刻印



• キーボードのキー番号



キー番号	刻 印	下 段	上 段 (Shift+)	Ctrl+	Alt+
1	~	29/60	29/7E	**/**	**/**, 29/00
2	1 !	1 02/31	! 02/21	**/**	78/00
3	2 @	2 03/32	@ 03/40	NUL 03/00	79/00
4	3 #	3 04/33	# 04/23	**/**	7A/00
5	4 \$	4 05/34	\$ 05/24	**/**	7B/00
6	5 %	5 06/35	% 06/25	**/**	7C/00
7	6 ^	6 07/36	^ 07/5E	RS 07/1E	7D/00
8	7 &	7 08/37	& 08/26	**/**	7E/00
9	8 *	8 09/38	* 09/2A	**/**	7F/00
10	9 (9 0A/39	(0A/28	**/**	80/00
11	0)	0 0B/30) 0B/29	**/**	81/00
12	- _	- 0C/2D	_ 0C/5F	US 0C/1F	82/00
13	= +	= 0D/3D	+ 0D/2B	**/**	83/00
14	(キーなし)				
15	Backspace	BS 0E/08	BS 0E/08	DEL 0E/7F	**/**, 0E/00
16	Tab Backtab	HT 0F/09	0F/00	**/**, 94/00	**/**, A5/00
17	Q	q 10/71	Q 10/51	DC1 10/11	10/00
18	W	w 11/77	W 11/57	ETB 11/17	11/00
19	E	e 12/65	E 12/45	ENQ 12/05	12/00
20	R	r 13/72	R 13/52	DC2 13/12	13/00
21	T	t 14/74	T 14/54	DC4 14/14	14/00
22	Y	y 15/79	Y 15/59	EM 15/19	15/00
23	U	u 16/75	U 16/55	NAK 16/15	16/00
24	I	i 17/69	I 17/49	HT 17/09	17/00
25	O	o 18/6F	O 18/4F	SI 18/0F	18/00
26	P	p 19/70	P 19/50	DLE 19/10	19/00
27	[{	[1A/5B	{ 1A/7B	ESC 1A/1B	**/**, 1A/00
28] }] 1B/5D	} 1B/7D	GS 1B/1D	**/**, 1B/00
29	\ ;	¥ 2B/5C	2B/7C	FS 2B/1C	**/**, 2B/00
30	CapsLock	**/**	**/**	**/**	**/**
31	A	a 1E/61	A 1E/41	SOH 1E/01	1E/00
32	S	s 1F/73	S 1F/53	DC3 1F/13	1F/00
33	D	d 20/64	D 20/44	EOT 20/04	20/00
34	F	f 21/66	F 21/46	ACK 21/06	21/00
35	G	g 22/67	G 22/47	BEL 22/07	22/00
36	H	h 23/68	H 23/48	BS 23/08	23/00
37	J	j 24/6A	J 24/4A	LF 24/0A	24/00
38	K	k 25/6B	K 25/4B	VT 25/0B	25/00
39	L	l 26/6C	L 26/4C	FF 26/0C	26/00
40	; :	; 27/3B	: 27/3A	**/**	**/**, 27/00
41	' "	' 28/27	" 28/22	**/**	**/**, 28/00
42	(キーなし)				
43	Enter	CR 1C/0D	CR 1C/0D	LF 1C/0A	**/**, 1C/00
44	Shift	**/**	**/**	**/**	**/**
45	(キーなし)				
46	Z	z 2C/7A	Z 2C/5A	SUB 2C/1A	2C/00
47	X	x 2D/78	X 2D/58	CAN 2D/18	2D/00
48	C	c 2E/63	C 2E/43	ETX 2E/03	2E/00
49	V	v 2F/76	V 2F/56	SYN 2F/16	2F/00
50	B	b 30/62	B 30/42	STX 30/02	30/00

51	N	n	31/6E	N	31/4E	SO	31/0E	31/00
52	M	m	32/6D	M	32/4D	CR	32/0D	32/00
53	, <	,	33/2C	<	33/3C		**/**	**/**, 33/00
54	. >	.	34/2E	>	34/3E		**/**	**/**, 34/00
55	/ ?	/	35/2F	?	35/3F		**/**	**/**, 35/00
56	(キーなし)							
57	Shift		**/**		**/**		**/**	**/**
58	Ctrl		**/**		**/**		**/**	**/**
59	(キーなし)							
60	Alt		**/**		**/**		**/**	**/**
61	Space	SP	39/20	SP	39/20	SP	39/20	SP 39/20
62	Alt		**/**		**/**		**/**	**/**
63	(キーなし)							
64	Ctrl		**/**		**/**		**/**	**/**
65	(キーなし)							
66	(キーなし)							
67	(キーなし)							
68	(キーなし)							
69	(キーなし)							
70	(キーなし)							
71	(キーなし)							
72	(キーなし)							
73	(キーなし)							
74	(キーなし)							
75	Insert		52/00, 52/E0		52/00, 52/E0		**/**, 92/E0	**/**, A2/00
76	Delete		53/00, 53/E0		53/00, 53/E0		**/**, 93/E0	**/**, A3/00
77	(キーなし)							
78	(キーなし)							
79	←		4B/00, 4B/E0		4B/00, 4B/E0		73/00, 73/E0	**/**, 9B/00
80	Home		47/00, 47/E0		47/00, 47/E0		77/00, 77/E0	**/**, 97/00
81	End		4F/00, 4F/E0		4F/00, 4F/E0		75/00, 75/E0	**/**, 9F/00
82	(キーなし)							
83	↑		48/00, 48/E0		48/00, 48/E0		**/**, 8D/E0	**/**, 98/00
84	↓		50/00, 50/E0		50/00, 50/E0		**/**, 91/E0	**/**, A0/00
85	PageUp		49/00, 49/E0		49/00, 49/E0		84/00, 84/E0	**/**, 99/00
86	PageDown		51/00, 51/E0		51/00, 51/E0		76/00, 76/E0	**/**, A1/00
87	(キーなし)							
88	(キーなし)							
89	→		4D/00, 4D/E0		4D/00, 4D/E0		74/00, 74/E0	**/**, 9D/00
90	NumLock		**/**		**/**		**/**	**/**
91	Home 7		47/00	7	47/37		77/00	Code Input
92	← 4		4B/00	4	4B/34		73/00	Code Input
93	End 1		4F/00	1	4F/31		75/00	Code Input
94	(キーなし)							
95	/	/	35/2F, E0/2F	/	35/2F, E0/2F		**/**, 95/00	**/**, A4/00
96	↑ 8		48/00	8	48/38		**/**, 8D/00	Code Input
97	5		**/**, 4C/00	5	4C/35		**/**, 8F/00	Code Input
98	↓ 2		50/00	2	50/32		**/**, 91/00	Code Input
99	Ins 0		52/00	0	52/30		**/**, 92/00	Code Input
100	* *	*	37/2A	*	37/2A		**/**, 96/00	**/**, 37/00
101	PgUp 9		49/00	9	49/39		84/00	Code Input

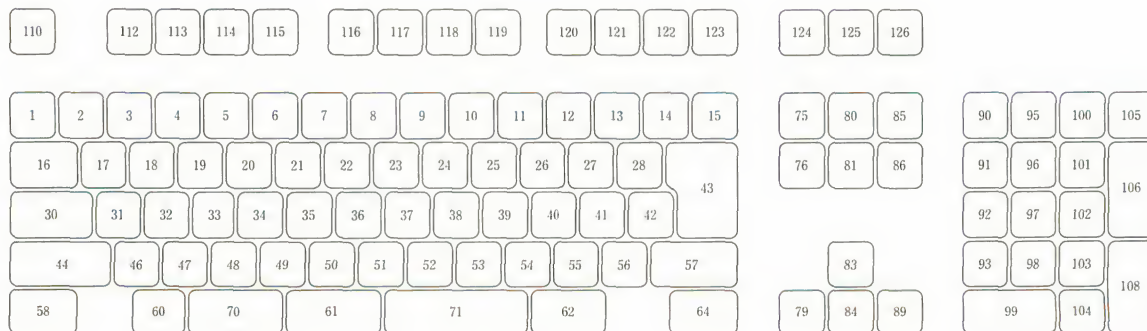
102	→ 6	4D/00	6 4D/36	74/00	Code Input
103	PgDn 3	51/00	3 51/33	76/00	Code Input
104	Del .	53/00	. 53/2E	**/**, 93/00	**/**
105	—	4A/2D	— 4A/2D	**/**, 8E/00	**/**, 4A/00
106	+ (キーなし)	+ 4E/2B	+ 4E/2B	**/**, 90/00	**/**, 4E/00
108	Enter (キーなし)	CR 1C/0D, E0/0D	CR 1C/0D, E0/0D	LF 1C/0A, E0/0A	**/**, A6/00
110	Esc (キーなし)	ESC 01/1B	ESC 01/1B	ESC 01/1B	**/**, 01/00
112	F1	3B/00	54/00	5E/00	68/00
113	F2	3C/00	55/00	5F/00	69/00
114	F3	3D/00	56/00	60/00	6A/00
115	F4	3E/00	57/00	61/00	6B/00
116	F5	3F/00	58/00	62/00	6C/00
117	F6	40/00	59/00	63/00	6D/00
118	F7	41/00	5A/00	64/00	6E/00
119	F8	42/00	5B/00	65/00	6F/00
120	F9	43/00	5C/00	66/00	70/00
121	F10	44/00	5D/00	67/00	71/00
122	F11	**/**, 85/00	**/**, 87/00	**/**, 89/00	**/**, 8B/00
123	F12	**/**, 86/00	**/**, 88/00	**/**, 8A/00	**/**, 8C/00
124	PrintScreen	**/**	**/**	72/00	**/**
125	ScrollLock	**/**	**/**	**/**	**/**
126	Pause	**/**	**/**	00/00	**/**
127	(キーなし)				
128	(キーなし)				
129	(キーなし)				
130	(キーなし)				
131	(キーなし)				
132	(キーなし)				
133	(キーなし)				

Appendix A-4 AX キーボード

・キーボードの刻印



・キーボードのキー番号



キー番号	刻 印	下 段	上 段 (Shift+)	Ctrl+	Alt+
1	Esc	ESC 01/1B	ESC 01/1B	ESC 01/1B	**/**, 01/00
2	1 ! ヌ	1 02/31	! 02/21	**/**	78/00
3	2 @ フ	2 03/32	@ 03/40	NUL 03/00	79/00
4	3 # ア ア	3 04/33	# 04/23	**/**	7A/00
5	4 \$ ウ ウ	4 05/34	\$ 05/24	**/**	7B/00
6	5 % エ エ	5 06/35	% 06/25	**/**	7C/00
7	6 ^ オ オ	6 07/36	^ 07/5E	RS 07/1E	7D/00
8	7 & ヤ ヤ	7 08/37	& 08/26	**/**	7E/00
9	8 * ユ ユ	8 09/38	* 09/2A	**/**	7F/00
10	9 (ヨ ヨ	9 0A/39	(0A/28	**/**	80/00
11	0) ワ ワ	0 0B/30) 0B/29	**/**	81/00
12	- _ ホ	- 0C/2D	- 0C/5F	US 0C/1F	82/00
13	= + ヘ	= 0D/3D	+ 0D/2B	**/**	83/00
14	¥ ー	¥ 2B/5C	2B/7C	FS 2B/1C	**/**, 2B/00
15	Backspace	BS 0E/08	BS 0E/08	DEL 0E/7F	**/**, 0E/00
16	Tab Backtab	HT 0F/09	0F/00	**/**, 94/00	**/**, A5/00
17	Q タ	q 10/71	Q 10/51	DC1 10/11	10/00
18	W テ	w 11/77	W 11/57	ETB 11/17	11/00
19	E イ イ	e 12/65	E 12/45	ENQ 12/05	12/00
20	R ス	r 13/72	R 13/52	DC2 13/12	13/00
21	T カ	t 14/74	T 14/54	DC4 14/14	14/00
22	Y ン	y 15/79	Y 15/59	EM 15/19	15/00
23	U ナ	u 16/75	U 16/55	NAK 16/15	16/00
24	I ニ	i 17/69	I 17/49	HT 17/09	17/00
25	O ラ	o 18/6F	O 18/4F	SI 18/0F	18/00
26	P セ	p 19/70	P 19/50	DLE 19/10	19/00
27	[{ " "	[1A/5B	{ 1A/7B	ESC 1A/1B	**/**, 1A/00
28] } ° ¨] 1B/5D	} 1B/7D	GS 1B/1D	**/**, 1B/00
29	(キーなし)				
30	Ctrl	**/**	**/**	**/**	**/**
31	A チ	a 1E/61	A 1E/41	SOH 1E/01	1E/00
32	S ト	s 1F/73	S 1F/53	DC3 1F/13	1F/00
33	D シ	d 20/64	D 20/44	EOT 20/04	20/00
34	F ハ	f 21/66	F 21/46	ACK 21/06	21/00
35	G キ	g 22/67	G 22/47	BEL 22/07	22/00
36	H ク	h 23/68	H 23/48	BS 23/08	23/00
37	J マ	j 24/6A	J 24/4A	LF 24/0A	24/00
38	K ノ	k 25/6B	K 25/4B	VT 25/0B	25/00
39	L リ	l 26/6C	L 26/4C	FF 26/0C	26/00
40	; : レ	; 27/3B	: 27/3A	**/**	**/**, 27/00
41	' " ケ	' 28/27	" 28/22	**/**	**/**, 28/00
42	` ~ ム ー	` 29/60	~ 29/7E	**/**	**/**, 29/00
43	Enter	CR 1C/0D	CR 1C/0D	LF 1C/0A	**/**, 1C/00
44	Shift	**/**	**/**	**/**	**/**
45	(キーなし)				
46	Z ツ ツ	z 2C/7A	Z 2C/5A	SUB 2C/1A	2C/00
47	X サ	x 2D/78	X 2D/58	CAN 2D/18	2D/00
48	C ソ	c 2E/63	C 2E/43	ETX 2E/03	2E/00
49	V ヒ	v 2F/76	V 2F/56	SYN 2F/16	2F/00
50	B コ	b 30/62	B 30/42	STX 30/02	30/00

51	N ミ	n	31/6E	N	31/4E	SO	31/0E	31/00
52	M モ	m	32/6D	M	32/4D	CR	32/0D	32/00
53	, < ネ	,	33/2C	<	33/3C		**/**	**/**, 33/00
54	. > ル	.	34/2E	>	34/3E		**/**	**/**, 34/00
55	/ ? メ	/	35/2F	?	35/3F		**/**	**/**, 35/00
56	\ ! ロ	¥	56/5C		56/7C	FS	56/1C	**/**, 56/00
57	Shift		**/**		**/**		**/**	**/**
58	CapsLock		**/**		**/**		**/**	**/**
59	(キーなし)							
60	Alt		**/**		**/**		**/**	**/**
61	Space	SP	39/20	SP	39/20	SP	39/20	SP 39/20
62	漢字		3A/00		3A/00		**/**	**/**
63	(キーなし)							
64	英数カナ		**/**		**/**		**/**	**/**
65	(キーなし)							
66	(キーなし)							
67	(キーなし)							
68	(キーなし)							
69	(キーなし)							
70	無変換		AB/00		AC/00		AD/00	AE/00
71	変換		A7/00		A8/00		A9/00	AA/00
72	(キーなし)							
73	(キーなし)							
74	(キーなし)							
75	Insert		52/00, 52/E0		52/00, 52/E0		**/**, 92/E0	**/**, A2/00
76	Delete		53/00, 53/E0		53/00, 53/E0		**/**, 93/E0	**/**, A3/00
77	(キーなし)							
78	(キーなし)							
79	←		4B/00, 4B/E0		4B/00, 4B/E0		73/00, 73/E0	**/**, 9B/00
80	Home		47/00, 47/E0		47/00, 47/E0		77/00, 77/E0	**/**, 97/00
81	End		4F/00, 4F/E0		4F/00, 4F/E0		75/00, 75/E0	**/**, 9F/00
82	(キーなし)							
83	↑		48/00, 48/E0		48/00, 48/E0		**/**, 8D/E0	**/**, 98/00
84	↓		50/00, 50/E0		50/00, 50/E0		**/**, 91/E0	**/**, A0/00
85	PageUp		49/00, 49/E0		49/00, 49/E0		84/00, 84/E0	**/**, 99/00
86	PageDown		51/00, 51/E0		51/00, 51/E0		76/00, 76/E0	**/**, A1/00
87	(キーなし)							
88	(キーなし)							
89	→		4D/00, 4D/E0		4D/00, 4D/E0		74/00, 74/E0	**/**, 9D/00
90	NumLock		**/**		**/**		**/**	**/**
91	Home 7		47/00	7	47/37		77/00	Code Input
92	← 4		4B/00	4	4B/34		73/00	Code Input
93	End 1		4F/00	1	4F/31		75/00	Code Input
94	(キーなし)							
95	/	/	35/2F, E0/2F	/	35/2F, E0/2F		**/**, 95/00	**/**, A4/00
96	↑ 8		48/00	8	48/38		**/**, 8D/00	Code Input
97	5		**/**, 4C/00	5	4C/35		**/**, 8F/00	Code Input
98	↓ 2		50/00	2	50/32		**/**, 91/00	Code Input
99	Ins 0		52/00	0	52/30		**/**, 92/00	Code Input
100	*	*	37/2A	*	37/2A		**/**, 96/00	**/**, 37/00
101	PgUp 9		49/00	9	49/39		84/00	Code Input

102	→	6	4D/00	6	4D/36	74/00	Code Input
103	PgDn	3	51/00	3	51/33	76/00	Code Input
104	Del	.	53/00	.	53/2E	**/**, 93/00	**/**
105		—	4A/2D	—	4A/2D	**/**, 8E/00	**/**, 4A/00
106		+	4E/2B	+	4E/2B	**/**, 90/00	**/**, 4E/00
107	(キーなし)						
108	Enter		CR 1C/0D, E0/0D	CR 1C/0D, E0/0D	LF 1C/0A, E0/0A	**/**, A6/00	
109	(キーなし)						
110	AX		D2/00	D3/00	D4/00	D5/00	
111	(キーなし)						
112	F1		3B/00	54/00	5E/00	68/00	
113	F2		3C/00	55/00	5F/00	69/00	
114	F3		3D/00	56/00	60/00	6A/00	
115	F4		3E/00	57/00	61/00	6B/00	
116	F5		3F/00	58/00	62/00	6C/00	
117	F6		40/00	59/00	63/00	6D/00	
118	F7		41/00	5A/00	64/00	6E/00	
119	F8		42/00	5B/00	65/00	6F/00	
120	F9		43/00	5C/00	66/00	70/00	
121	F10		44/00	5D/00	67/00	71/00	
122	F11		**/**, 85/00	**/**, 87/00	**/**, 89/00	**/**, 8B/00	
123	F12		**/**, 86/00	**/**, 88/00	**/**, 8A/00	**/**, 8C/00	
124	PrintScreen		**/**	**/**	72/00	**/**	
125	ScrollLock		**/**	**/**	**/**	**/**	
126	Pause		**/**	**/**	00/00	**/**	
127	(キーなし)						
128	(キーなし)						
129	(キーなし)						
130	(キーなし)						
131	(キーなし)						
132	(キーなし)						
133	(キーなし)						

Appendix A-5 東芝 J-3100 キーボード

・キーボードの配列



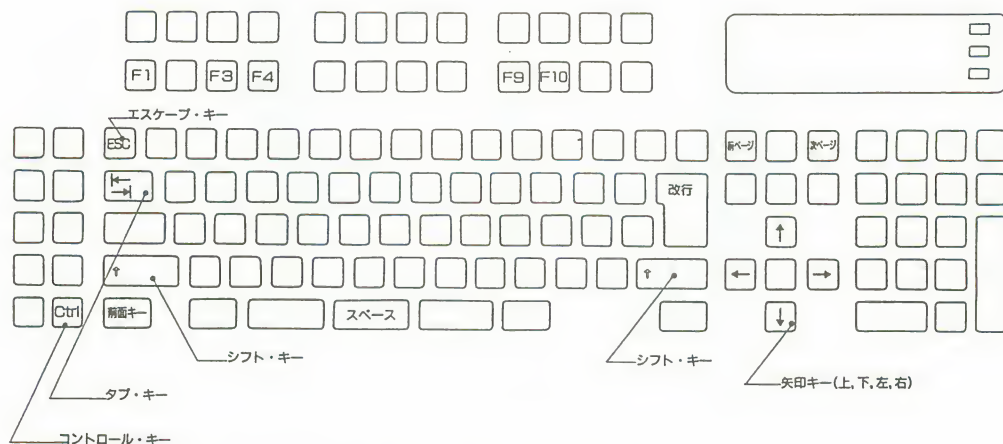
キー番号	刻 印	下 段	上 段 (Shift+)	Ctrl+	Alt+
1	Esc	ESC 01/1B	ESC 01/1B	ESC 01/1B	*/*/*, 01/00
2	1 ! め	1 02/31	! 02/21	*/*/*	78/00
3	2 @ ふ	2 03/32	@ 03/40	NUL 03/00	79/00
4	3 # あ あ	3 04/33	# 04/23	*/*/*	7A/00
5	4 \$ う う	4 05/34	\$ 05/24	*/*/*	7B/00
6	5 % え え	5 06/35	% 06/25	*/*/*	7C/00
7	6 ^ お お	6 07/36	^ 07/5E	RS 07/1E	7D/00
8	7 & や や	7 08/37	& 08/26	*/*/*	7E/00
9	8 * ゆ ゆ	8 09/38	* 09/2A	*/*/*	7F/00
10	9 (よ よ	9 0A/39	(0A/28	*/*/*	80/00
11	0) わ	0 0B/30) 0 B/29	*/*/*	81/00
12	- _ ほ	- 0C/2D	- 0C/5F	US 0C/1F	82/00
13	= + ヘ	= 0D/3D	+ 0D/2B	*/*/*	83/00
14	¥ ー	¥ 55/5C	55/7C	FS 55/1C	*/*/*, 2B/00
15	Backspace	BS 0E/08	BS 0E/08	DEL 0E/7F	*/*/*, 0E/00
16	Tab	HT 0F/09	0F/00	*/*/*, 94/00	*/*/*, A5/00
17	Q た	q 10/71	Q 10/51	DC1 10/11	10/00
18	W て	w 11/77	W 11/57	ETB 11/17	11/00
19	E い い	e 12/65	E 12/45	ENQ 12/05	12/00
20	R す	r 13/72	R 13/52	DC2 13/12	13/00
21	T か	t 14/74	T 14/54	DC4 14/14	14/00
22	Y ん	y 15/79	Y 15/59	EM 15/19	15/00
23	U な	u 16/75	U 16/55	NAK 16/15	16/00
24	I に	i 17/69	I 17/49	HT 17/09	17/00
25	O ら	o 18/6F	O 18/4F	SI 18/0F	18/00
26	P せ	p 19/70	P 19/50	DLE 19/10	19/00
27	[{ “	[1A/5B	{ 1A/7B	GS 1A/1B	*/*/*, 1A/00
28] } ° 「] 1B/5D	} 1B/7D	FS 1B/1D	*/*/*, 1B/00
29	(キーなし)				
30	Ctrl	*/*/*	*/*/*	*/*/*	*/*/*
31	A ち	a 1E/61	A 1E/41	SOH 1E/01	1E/00
32	S と	s 1F/73	S 1F/53	DC3 1F/13	1F/00
33	D し	d 20/64	D 20/44	EOT 20/04	20/00
34	F は	f 21/66	F 21/46	ACK 21/06	21/00
35	G き	g 22/67	G 22/47	BEL 22/07	22/00
36	H く	h 23/68	H 23/48	BS 23/08	23/00
37	J ま	j 24/6A	J 24/4A	LF 24/0A	24/00
38	K の	k 25/6B	K 25/4B	VT 25/0B	25/00
39	L り	l 26/6C	L 26/4C	FF 26/0C	26/00
40	; : れ	; 27/3B	: 27/3A	*/*/*	*/*/*, 27/00
41	' " け	' 28/27	" 28/22	*/*/*	*/*/*, 28/00
42	~ む 「	~ 29/60	~ 29/7E	*/*/*	*/*/*, 29/00
43	Enter	CR 1C/0D	CR 1C/0D	LF 1C/0A	*/*/*, 1C/00
44	Shift	*/*/*	*/*/*	*/*/*	*/*/*
45	(キーなし)				
46	Z つ っ	z 2C/7A	Z 2C/5A	SUB 2C/1A	2C/00
47	X さ	x 2D/78	X 2D/58	CAN 2D/18	2D/00
48	C そ	c 2E/63	C 2E/43	ETX 2E/03	2E/00
49	V ひ	v 2F/76	V 2F/56	SYN 2F/16	2F/00
50	B こ	b 30/62	B 30/42	STX 30/02	30/00

51	N み	n	31/6E	N	31/4E	SO	31/0E	31/00
52	M も	m	32/6D	M	32/4D	CR	32/0D	32/00
53	, < ね	,	33/2C	<	33/3C		**/**	**/**, 33/00
54	. > る	.	34/2E	>	34/3E		**/**	**/**, 34/00
55	/ ? め	/	35/2F	?	35/3F		**/**	**/**, 35/00
56	\ : ろ	¥	2B/5C		2B/7C	FS	2B/1C	2B/00
57	Shift		**/**		**/**		**/**	**/**
58	CapsLock		**/**		**/**		**/**	**/**
59	(キーなし)							
60	Alt		**/**		**/**		**/**	**/**
61	Space	SP	39/20	SP	39/20	SP	39/20	SP 39/20
62	漢字		3A/00		A8/00		AA/00	B5/00
63	(キーなし)							
64	カナ		B6/00		B3/00		B6/00	B9/00
65	(キーなし)							
66	(キーなし)							
67	(キーなし)							
68	(キーなし)							
69	(キーなし)							
70	(キーなし)							
71	(キーなし)							
72	(キーなし)							
73	(キーなし)							
74	(キーなし)							
75	Insert		52/00, 52/E0		52/00, 52/E0		**/**, 92/E0	**/**, A2/00
76	Delete		53/00, 53/E0		53/00, 53/E0		**/**, 93/E0	**/**, A3/00
77	(キーなし)							
78	(キーなし)							
79	←		4B/00, 4B/E0		4B/00, 4B/E0		73/00, 73/E0	**/**, 9B/00
80	Home		47/00, 47/E0		47/00, 47/E0		77/00, 77/E0	**/**, 97/00
81	End		4F/00, 4F/E0		4F/00, 4F/E0		75/00, 75/E0	**/**, 9F/00
82	(キーなし)							
83	↑		48/00, 48/E0		48/00, 48/E0		**/**, 8D/E0	**/**, 98/00
84	↓		50/00, 50/E0		50/00, 50/E0		**/**, 91/E0	**/**, A0/00
85	PageUp		49/00, 49/E0		49/00, 49/E0		84/00, 84/E0	**/**, 99/00
86	PageDown		51/00, 51/E0		51/00, 51/E0		76/00, 76/E0	**/**, A1/00
87	(キーなし)							
88	(キーなし)							
89	→		4D/00, 4D/E0		4D/00, 4D/E0		74/00, 74/E0	**/**, 9D/00
90	NumLock		**/**		**/**		**/**	**/**
91	Home 7		47/00	7	47/37		77/00	Code Input
92	← 4		4B/00	4	4B/34		73/00	Code Input
93	End 1		4F/00	1	4F/31		75/00	Code Input
94	(キーなし)							
95	/	/	35/2F, E0/2F	/	35/2F, E0/2F		**/**, 95/00	**/**, A4/00
96	↑ 8		48/00	8	48/38		**/**, 8D/00	Code Input
97	5		**/**, 4C/00	5	4C/35		**/**, 8F/00	Code Input
98	↓ 2		50/00	2	50/32		**/**, 91/00	Code Input
99	Ins 0		52/00	0	52/30		**/**, 92/00	Code Input
100	* *	*	37/2A	*	37/2A		**/**, 96/00	**/**, 37/00
101	PgUp 9		49/00	9	49/39		84/00	Code Input

102	→	6	4D/00	6	4D/36	74/00	Code Input
103	PgDn	3	51/00	3	51/33	76/00	Code Input
104	Del	.	53/00	.	53/2E	**/**, 93/00	**/**
105		—	— 4A/2D	—	4A/2D	**/**, 8E/00	**/**, 4A/00
106		+	— 4E/2B	+	4E/2B	**/**, 90/00	**/**, 4E/00
107	(キーなし)						
108	Enter		CR 1C/0D, E0/0D	CR 1C/0D, E0/0D	LF 1C/0A, E0/0A		**/**, A6/00
109	(キーなし)						
110			AF/00	D3/00	B2/00		B2/00
111	(キーなし)						
112	F1		3B/00	54/00	5E/00		68/00
113	F2		3C/00	55/00	5F/00		69/00
114	F3		3D/00	56/00	60/00		6A/00
115	F4		3E/00	57/00	61/00		6B/00
116	F5		3F/00	58/00	62/00		6C/00
117	F6		40/00	59/00	63/00		6D/00
118	F7		41/00	5A/00	64/00		6E/00
119	F8		42/00	5B/00	65/00		6F/00
120	F9		43/00	5C/00	66/00		70/00
121	F10		44/00	5D/00	67/00		71/00
122	F11		**/**, 85/00	**/**, 87/00	**/**, 89/00		**/**, 8B/00
123	F12		**/**, 86/00	**/**, 88/00	**/**, 8A/00		**/**, 8C/00
124	PrintScreen		**/**	**/**	72/00		**/**
125	ScrollLock		**/**	**/**	**/**		**/**
126	Pause		**/**	**/**	00/00		**/**
127	(キーなし)						
128	(キーなし)						
129	(キーなし)						
130	(キーなし)						
131	(キーなし)						
132	(キーなし)						
133	(キーなし)						

Appendix A-6 DOS/V 非公式対応のキーボード

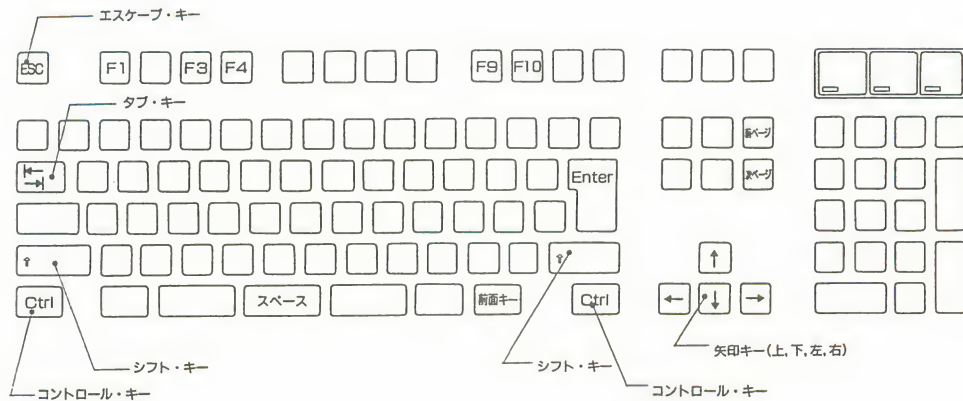
・5576-001 型キーボード



122	123	124	125	126	127	128	129	130	131	132	133
110	111	112	113	114	115	116	117	118	119	120	121

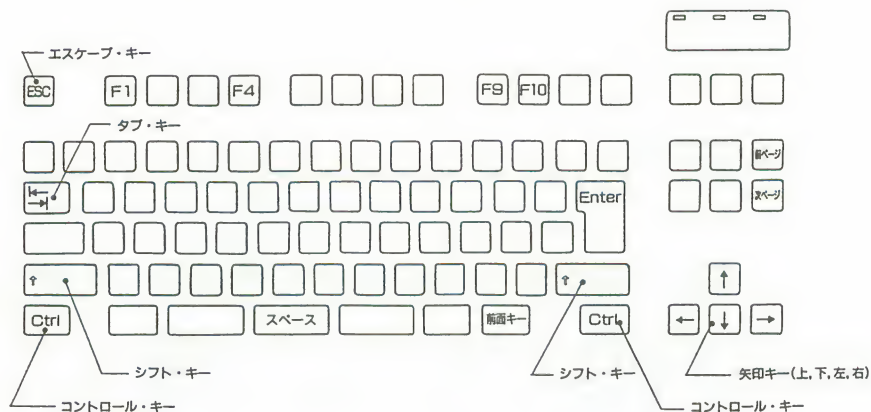
70	65	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	75	80	85	90	95	100	105
71	66	16	17	18	19	20	21	22	23	24	25	26	27	28	29		76	81	86	91	96	101	106
72	67	30	31	32	33	34	35	36	37	38	39	40	41	42				82		92	97	102	
73	68	44		46	47	48	49	50	51	52	53	54	55	56	57		78	83	88	93	98	103	107
74	69	58		59	60		61		62	63				64			84			99	104		

・ 5576-002 型キーボード



110	112 113 114 115				116 117 118 119				120 121 122 123				124 125 126								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	75	80	85	90	95	100	105
16	17	18	19	20	21	22	23	24	25	26	27	28	43		76	81	86	91	96	101	106
30	31	32	33	34	35	36	37	38	39	40	41	42						92	97	102	
44			46	47	48	49	50	51	52	53	54	55	56	57	83			93	98	103	108
58			60	131		61		132		133		62			64	79	84	89	99		104

・ 5576-003 型キーボード(5535・5523 も同系列)



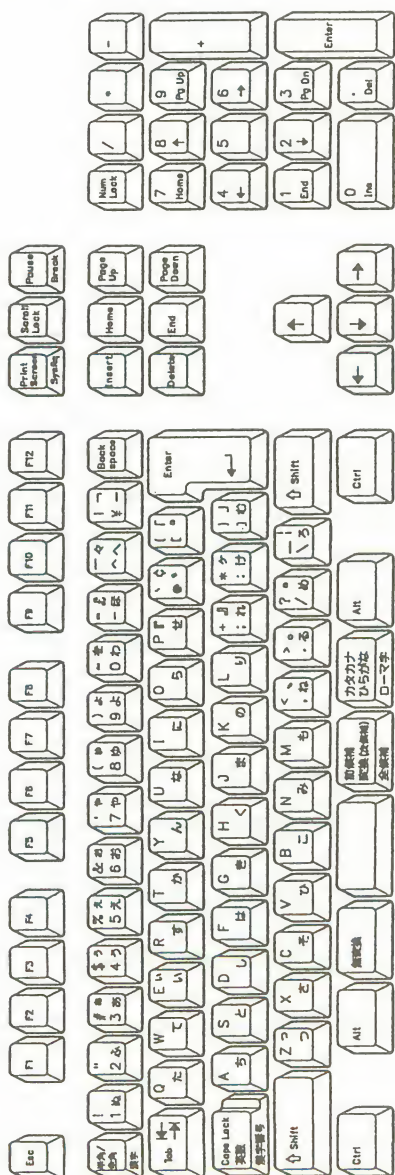
110		112 113 114 115				116 117 118 119				120 121 122 123				124 125 126			数値キー・パッド (17キー)				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	75	80	85	90	95	100	105
16	17	18	19	20	21	22	23	24	25	26	27	28	43		76	81	86	91	96	101	106
30	31	32	33	34	35	36	37	38	39	40	41	42			92	97	102				
44		46	47	48	49	50	51	52	53	54	55	56	57		83			93	98	103	108
58		60	131		61	132	133	62			64		79	84	89	99	104				

数値キー・パッドがついていない場合で、Num Lock 状態のとき、網かけ部分のキーが数値キー・パッド (Num Lock 状態) の代用として同等の働きをします。

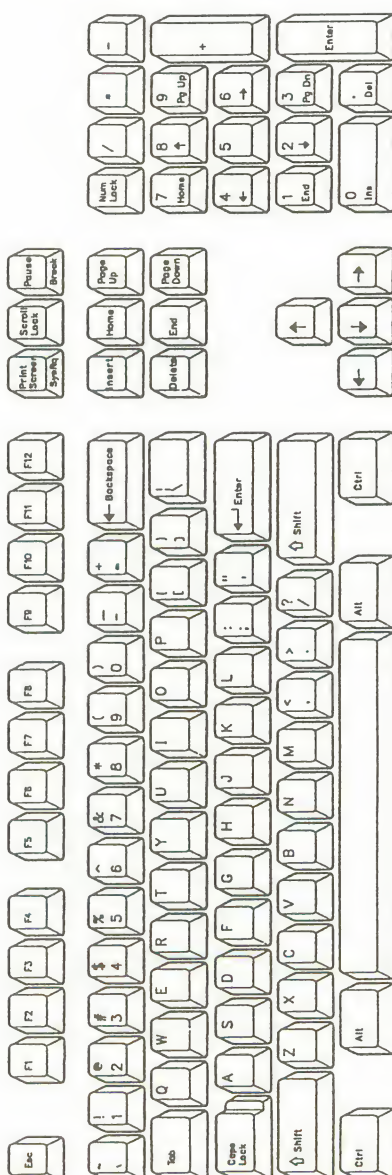
キー番号	対応する数値キー・パッドの番号	キー番号	対応する数値キー・パッドの番号
8	91	38	98
9	95	39	103
10	100	40	106
12	105	41	90
23	92	52	99
24	97	53	100
25	102	54	104
27	108	55	95
37	93		

Appendix A-7 世界各国のキーボード

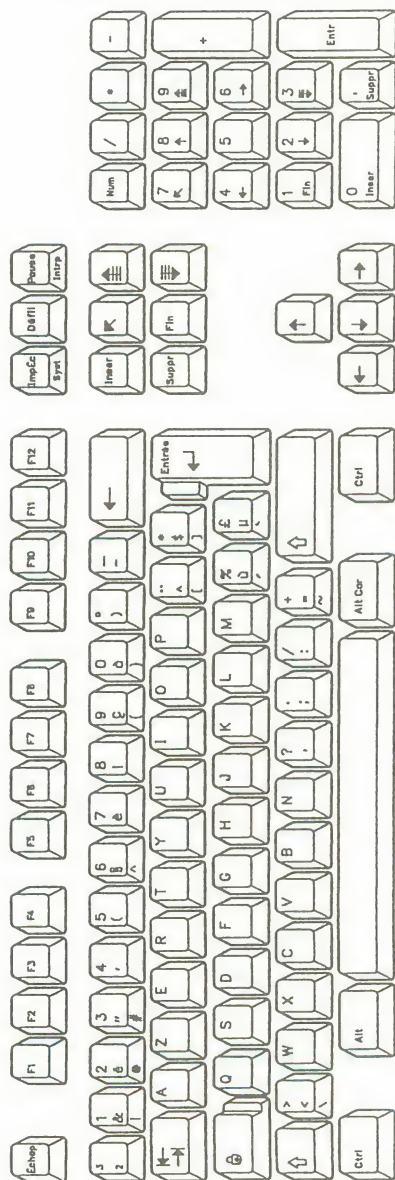
・ 日 本



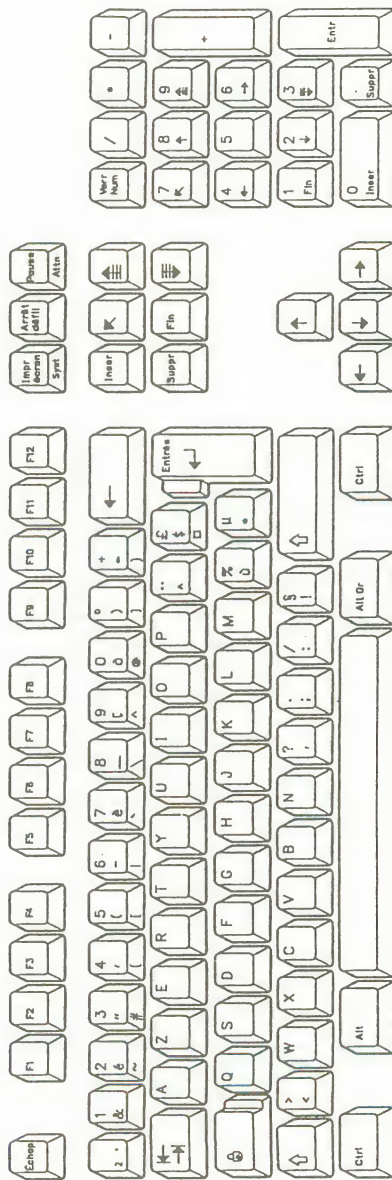
・ アメリカ合衆国



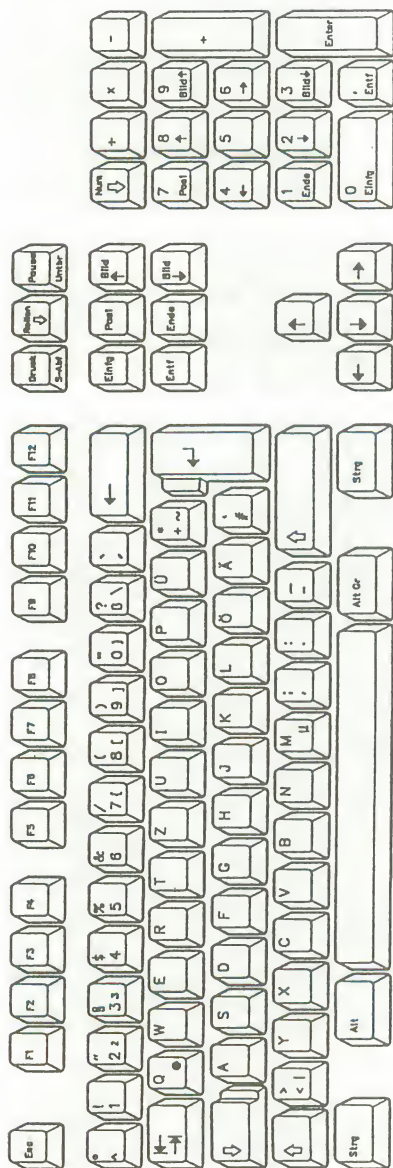
・ フランス(120)



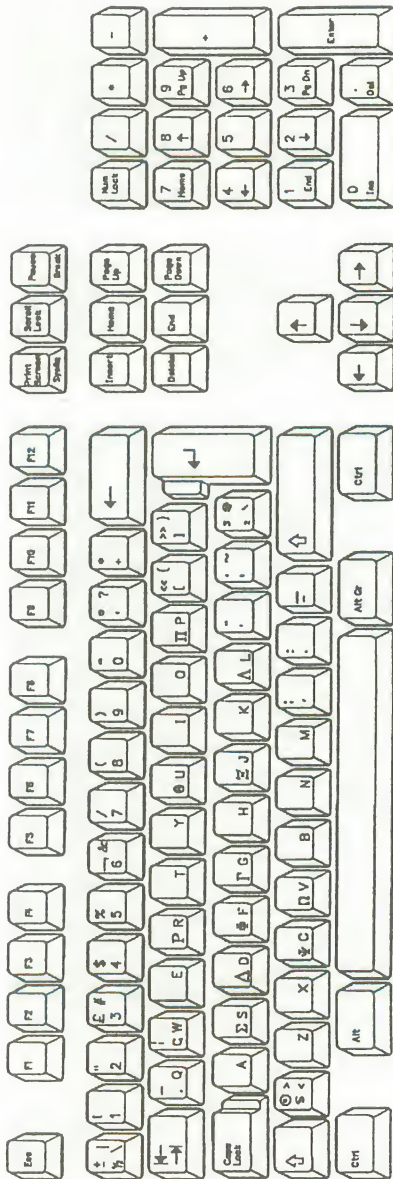
・ フランス(189)



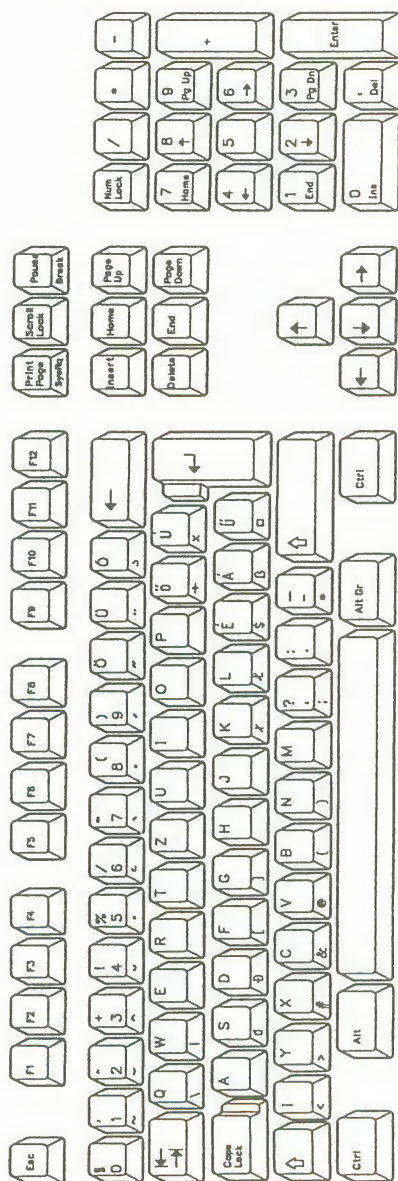
• ドイツ



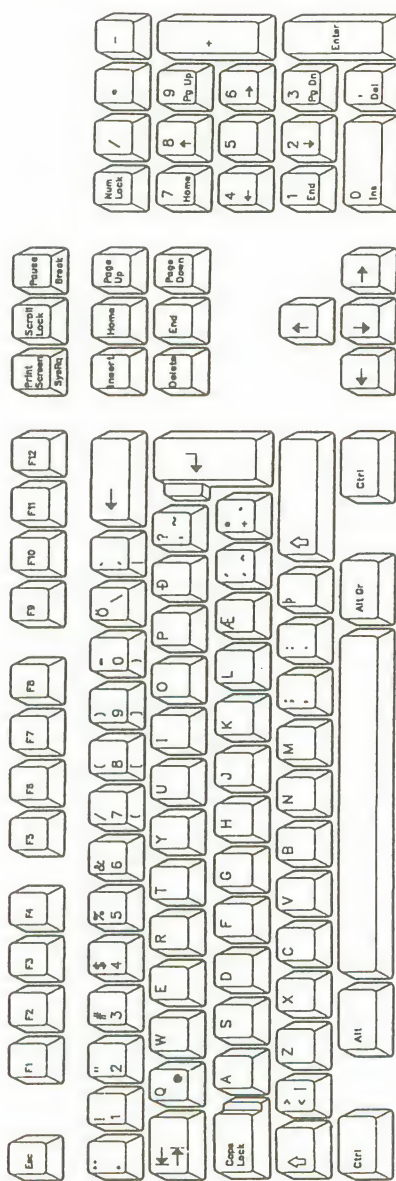
• ギリシャ



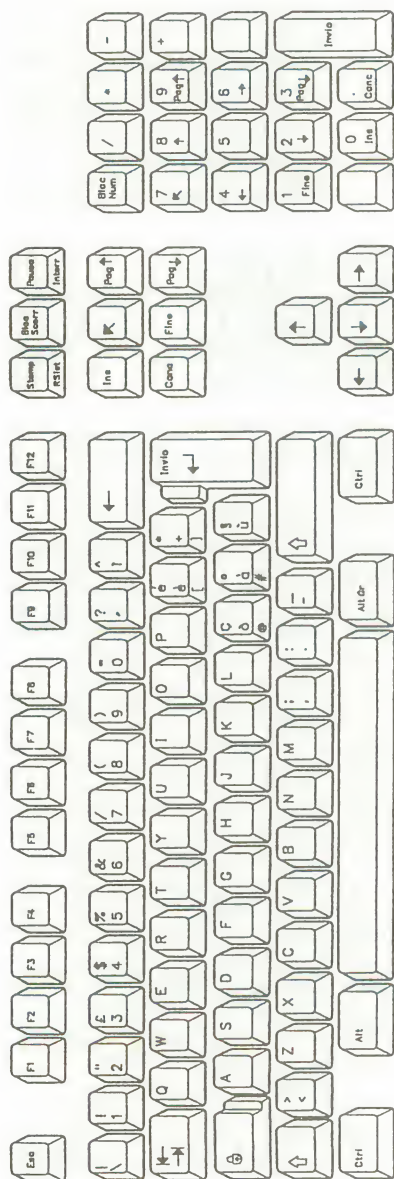
・ハンガリー



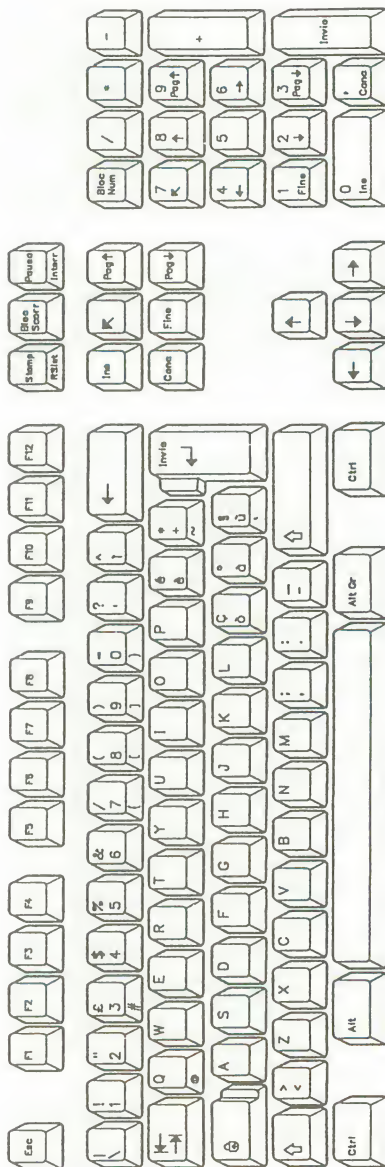
・アイスランド



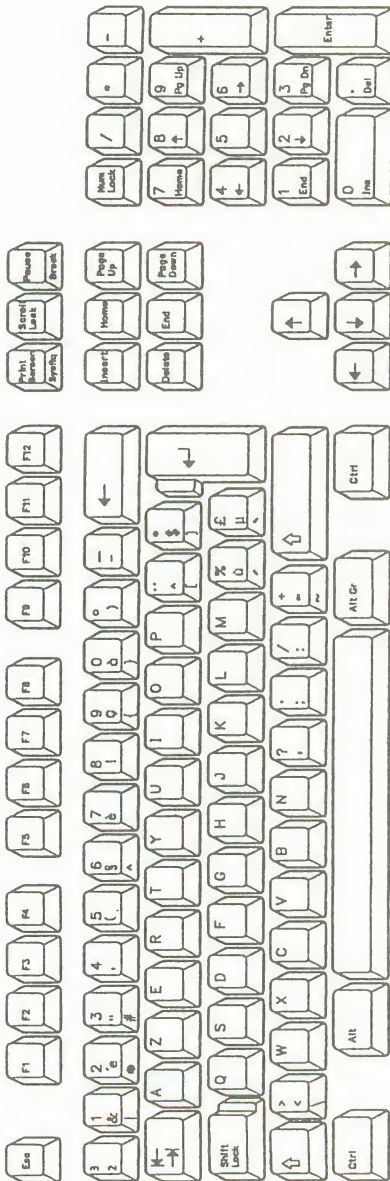
・イタリア(141)



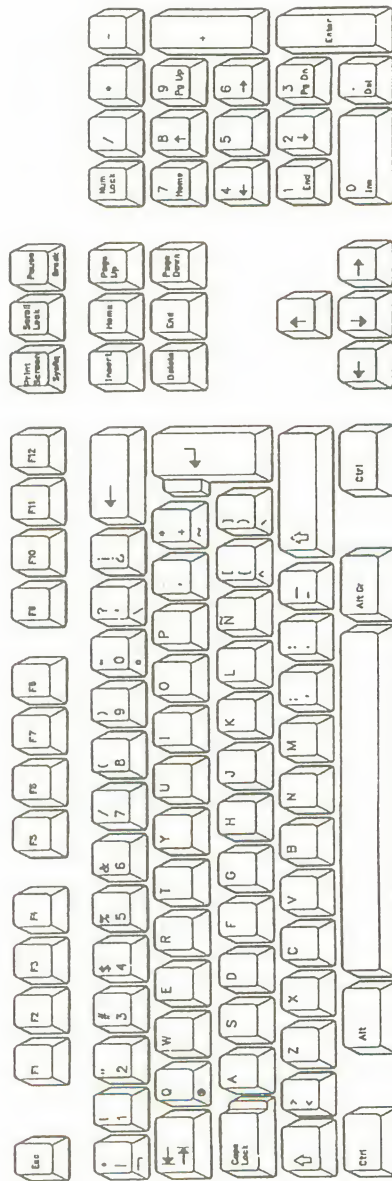
・イタリア(142)



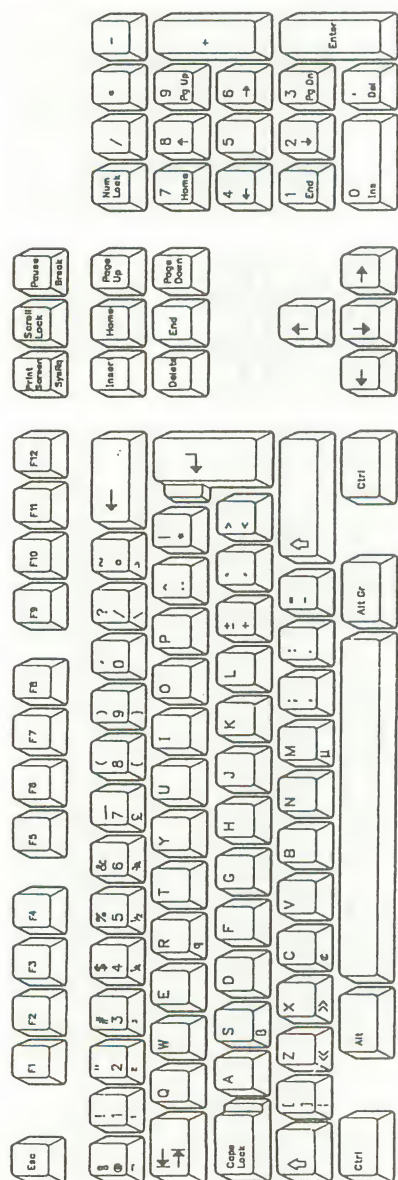
・ベルギー



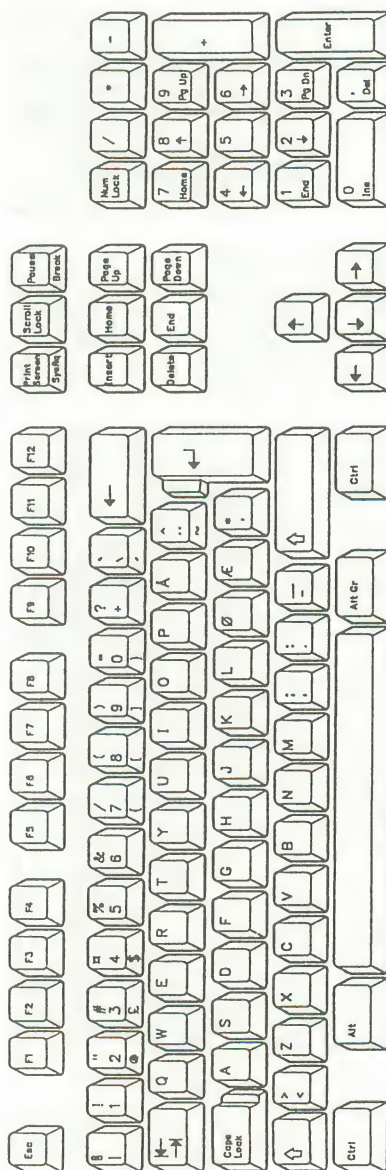
・ラテン・アメリカ



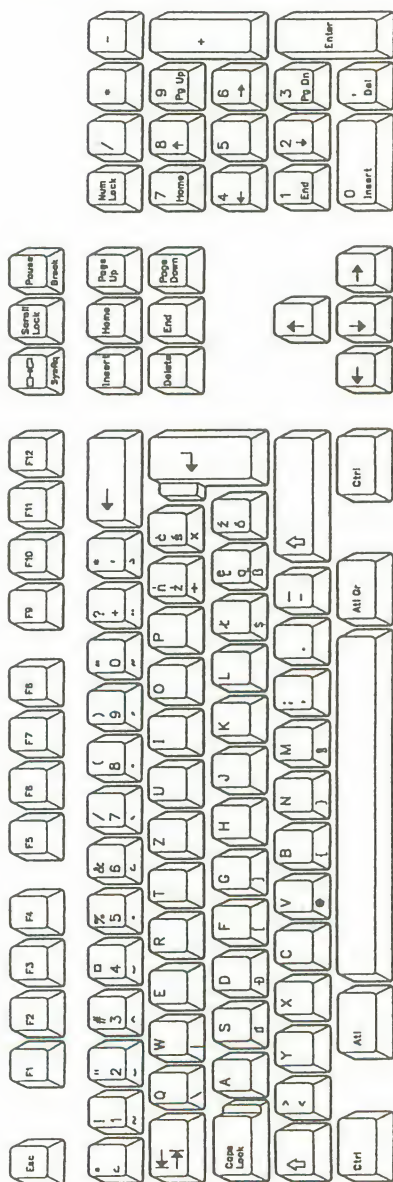
• オランダ



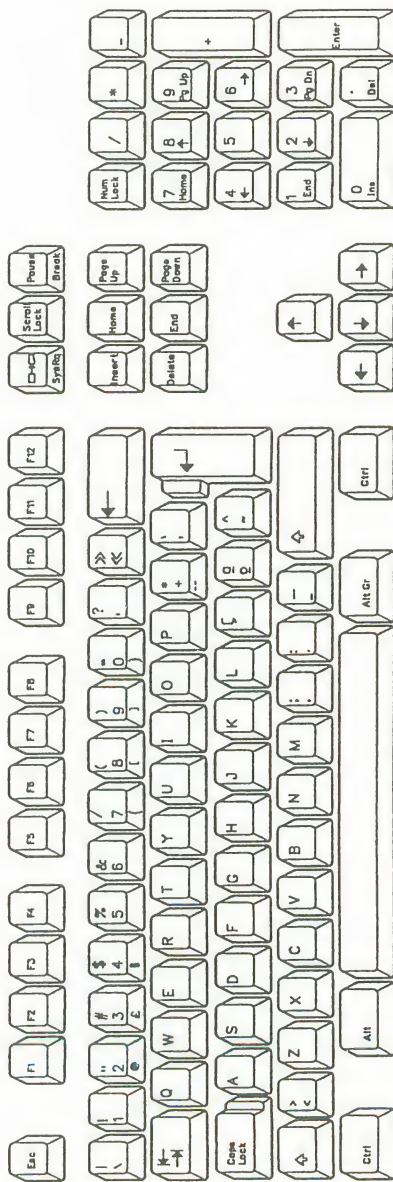
• ノルウェー



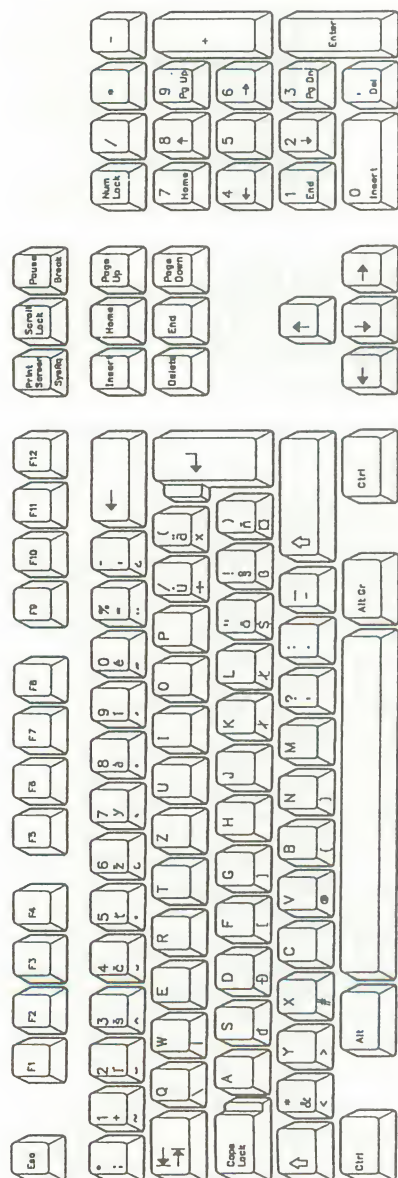
・ポーランド



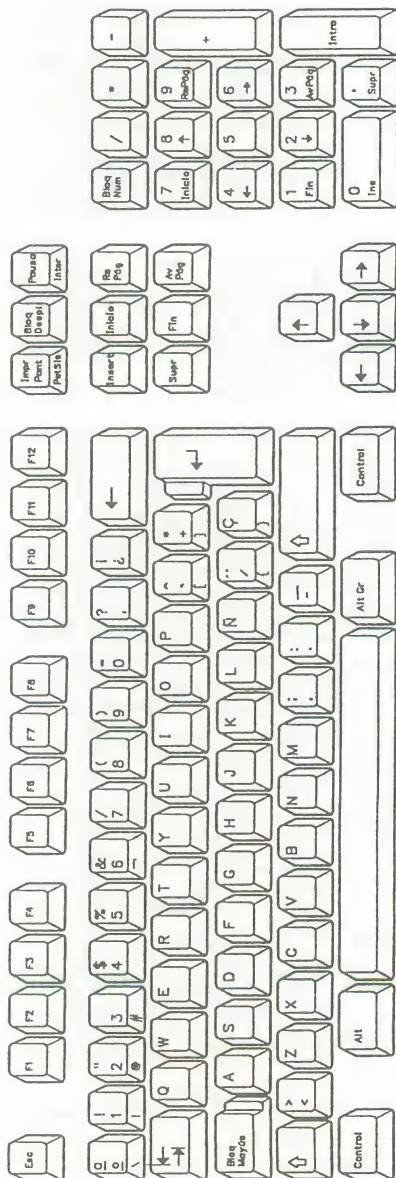
・ポルトガル



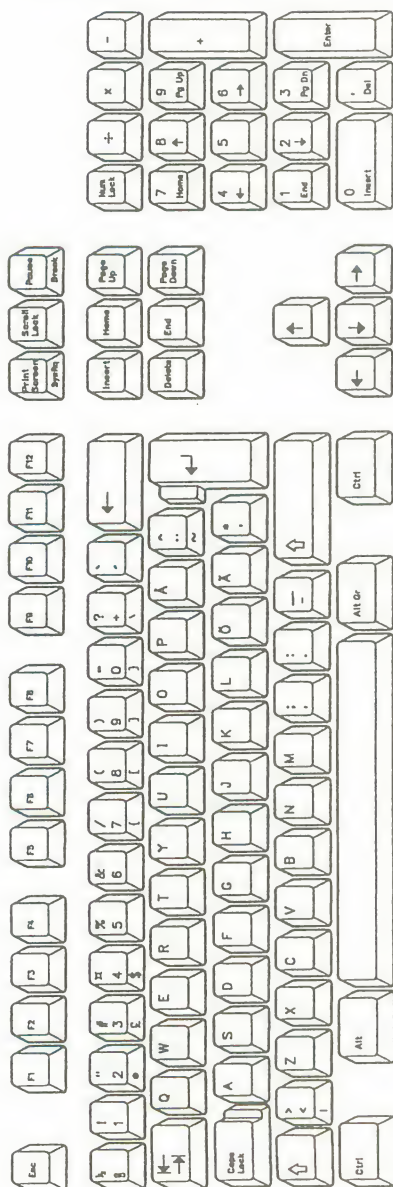
・チェコスロバキア(スロバキア語圏)



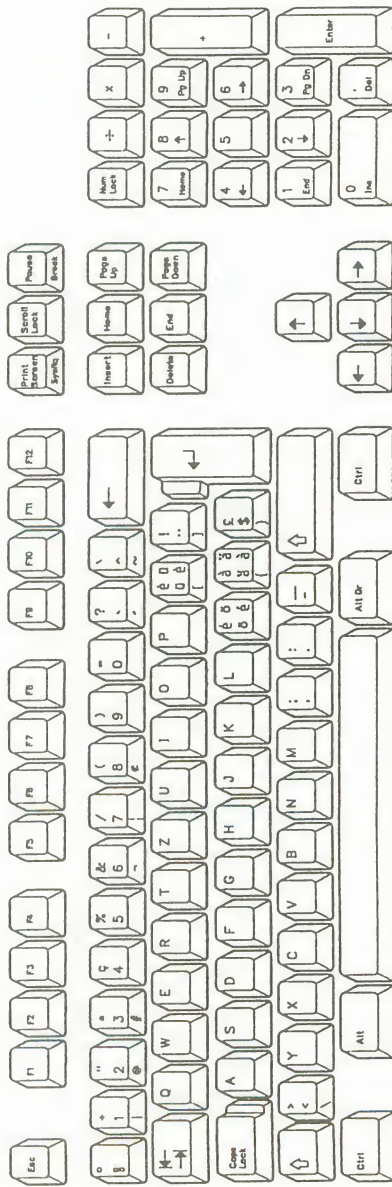
・スペイン



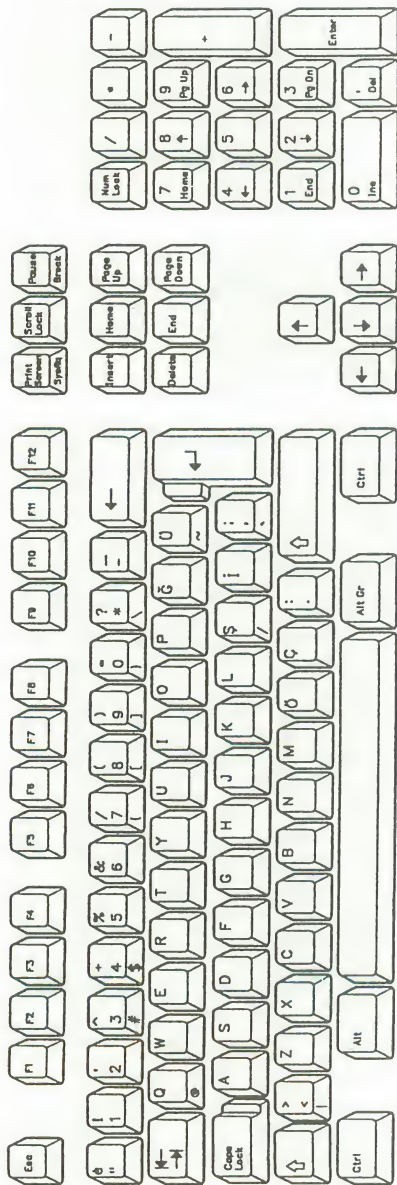
・ スウェーデン/フィンランド



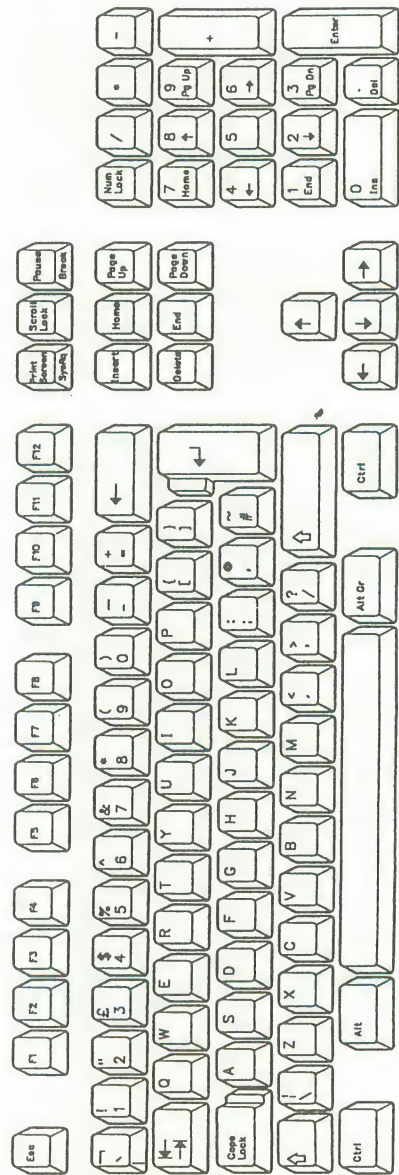
・ スイス(フランス語圏/ドイツ語圏)



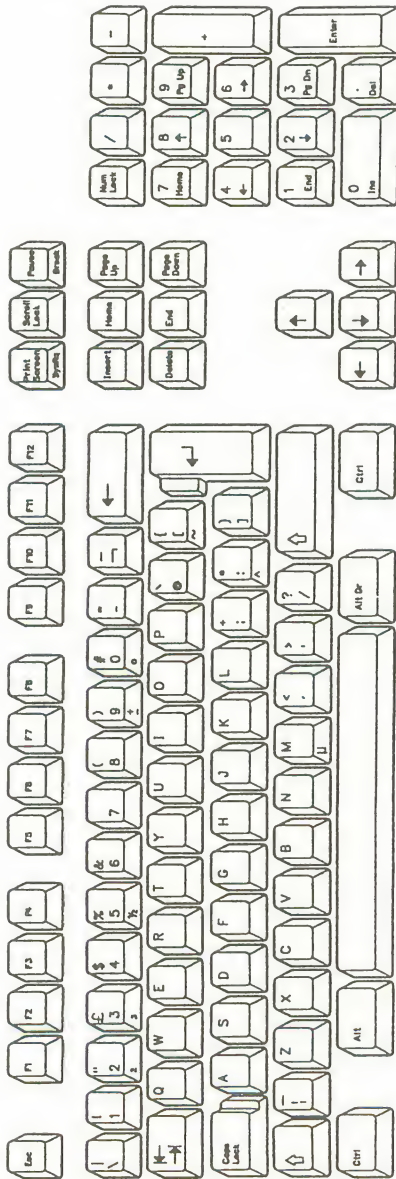
• トルコ



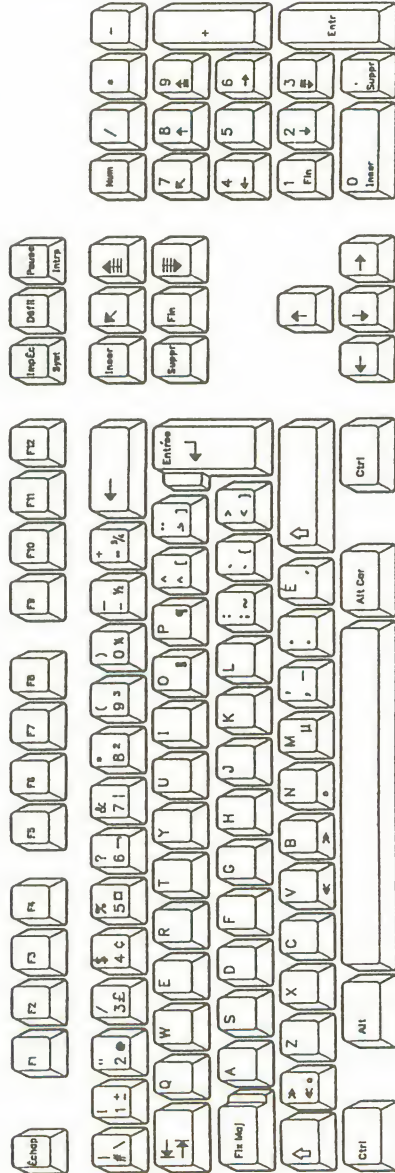
• イギリス(166)



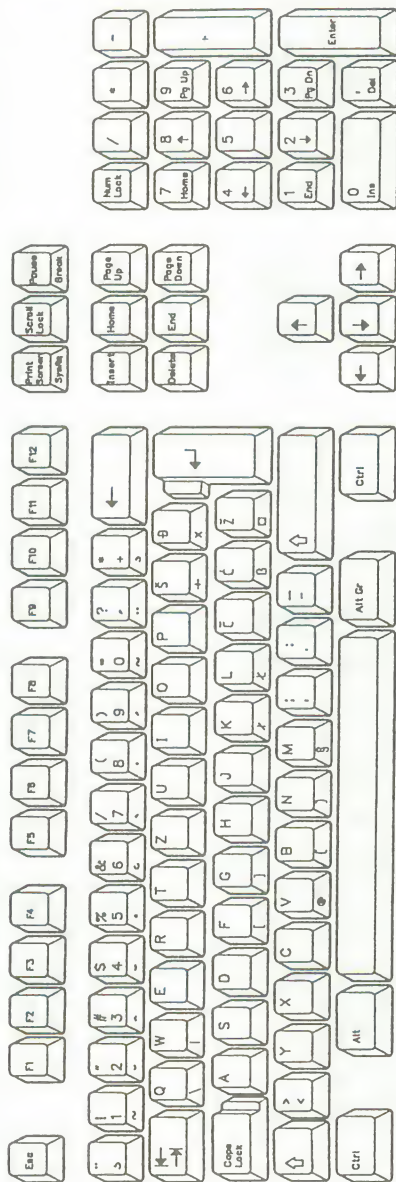
・ イギリス (168)



・ カナダ



• ユーゴスラビア



Appendix A-8 ANSI エスケープ・シーケンス一覧

ANSI エスケープ・シーケンスは米国で標準規格として定められた、コンソール制御規格です。この規格はどの装置でも共通のもので、互換性を考慮するときには重要な制御方式となります。ただし DOS/V の場合は、起動時にドライバ「ANSI.SYS」を組み込んでおく必要があります。

カーソル位置の指定

ESC [行位置 ; 桁位置 H または ESC [行位置 ; 桁位置 f

カーソルを指定の位置へ移動させます。位置を指定しなければホーム・ポジション(行=1, 桁=1)へ移動します。

カーソル位置の上移動

ESC [行数 A

カーソルを指定行数だけ上へ移動します。最上行では無視されます。

カーソル位置の下移動

ESC [行数 B

カーソルを指定行数だけ下へ移動します。最下行では無視されます。

カーソル位置の右移動

ESC [行数 C

カーソルを指定桁数だけ右へ移動します。最右桁では無視されます。

カーソル位置の左移動

ESC [行数 D

カーソルを指定桁数だけ左へ移動します。最左桁では無視されます。

カーソル位置の保存

ESC [s

現在のカーソル位置を保存します。

カーソル位置の復元

ESC [u

保存されたカーソル位置を復元します。

画面の消去

ESC [2J

画面を消去し、カーソルをホーム・ポジション(行=1, 桁=1)へ移動します。

行の消去

ESC [K

現在のカーソル位置から、行の終端までを消去します。

文字属性の設定

ESC [属性;…属性 m

文字属性を設定します。設定は複数可能で、設定された属性は次に設定が行われるまで有効です。

0：属性を無効にする
 1：高輝度またはボールド
 2：アンダーライン
 7：リバーズ
 8：非表示

30：黒の文字色
 31：赤の文字色
 32：緑の文字色
 33：黄色の文字色
 34：青の文字色
 35：紫の文字色
 36：水色の文字色
 37：白の文字色

40 黒の背景色
 41 赤の背景色
 42 緑の背景色
 43 黄色の背景色
 44 青の背景色
 45 紫の背景色
 46 水色の背景色
 47 白の背景色

画面モードの設定

ESC [=画面モード h

画面モードを設定します。

3	80 文字×25 行	カラー(テキスト)
17	640×480 ドット	モノクロ(2 色グラフィック)
18	640×480 ドット	カラー(16 色グラフィック)
114	640×480 ドット	カラー(16 色 80 文字×25 行)
115	80 文字×25 行	カラー(拡張テキスト)

キーの再割り当て

ESC [コード;文字列 p



キーを特定の文字列として、再定義します。

(次ページの一覧表参照)

・キーの再割り当て一覧表

キー	コード	↑ + コード	Ctrl + コード	Alt + コード
F1	0;59	0;84	0;94	0;104
F2	0;60	0;85	0;95	0;105
F3	0;61	0;86	0;96	0;106
F4	0;62	0;87	0;97	0;107
F5	0;63	0;88	0;98	0;108
F6	0;64	0;89	0;99	0;109
F7	0;65	0;90	0;100	0;110
F8	0;66	0;91	0;101	0;111
F9	0;67	0;92	0;102	0;112
F10	0;68	0;93	0;103	0;113
F11	(0;133)	(0;135)	(0;137)	(0;139)
F12	(0;134)	(0;136)	(0;138)	(0;140)
HOME (数値キー)	0;71	55	0;119	---
↑ (数値キー)	0;72	56	(0;141)	---
Page Up (数値キー)	0;73	57	0;132	---
← (数値キー)	0;75	52	0;115	---
→ (数値キー)	0;77	54	0;116	---
End (数値キー)	0;79	49	0;117	---
↓ (数値キー)	0;80	50	(0;145)	---
Page Down (数値キー)	0;81	51	0;118	---
Insert (数値キー)	0;82	48	(0;146)	---
Delete (数値キー)	0;83	46	(0;147)	---
Home	0;71(224;71)	0;71(224;71)	0;119(224;119)	(224;151)
↑	0;72(224;72)	0;72(224;72)	(224;141)	(224;152)
PgUp	0;73(224;73)	0;73(224;73)	0;132(224;132)	(224;153)
←	0;75(224;75)	0;75(224;75)	0;115(224;115)	(224;155)
→	0;77(224;77)	0;77(224;77)	0;116(224;116)	(224;157)
End	0;79(224;79)	0;79(224;79)	0;117(224;117)	(224;159)
↓	0;80(224;80)	0;80(224;80)	(224;145)	(224;154)
PgDn	0;81(224;81)	0;81(224;81)	0;118(224;118)	(224;161)
Ins	0;82(224;82)	0;82(224;82)	(224;146)	(224;162)
Del	0;83(224;83)	0;83(224;83)	(224;147)	(224;163)
Print Screen	---	---	0;114	---
Pause	---	---	0;0	---
Back Space	8	8	127	(0)
Enter	13	13	10	(0;28)
Tab	9	0;15	(0;148)	(0;165)
Esc	27	27	27	(0;01)
A	97	65	1	0;30
B	98	66	2	0;48

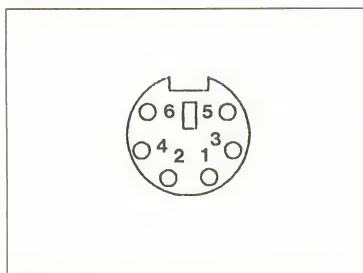
C	99	67	3	0; 46
D	100	68	4	0; 32
E	101	69	5	0; 18
F	102	70	6	0; 33
G	103	71	7	0; 34
H	104	72	8	0; 35
I	105	73	9	0; 23
J	106	74	10	0; 36
K	107	75	11	0; 37
L	108	76	12	0; 38
M	109	77	13	0; 50
N	110	78	14	0; 49
O	111	79	15	0; 24
P	112	80	16	0; 25
Q	113	81	17	0; 16
R	114	82	18	0; 19
S	115	83	19	0; 31
T	116	84	20	0; 20
U	117	85	21	0; 22
V	118	86	22	0; 47
W	119	87	23	0; 17
X	120	88	24	0; 45
Y	121	89	25	0; 21
Z	122	90	26	0; 44
1	49	33	---	0; 120
2	50	34	0; 3	0; 121
3	51	35	---	0; 122
4	52	36	---	0; 123
5	53	37	---	0; 124
6	54	38	30	0; 125
7	55	39	---	0; 126
8	56	40	---	0; 126
9	57	41	---	0; 127
0	48	(0; 11)	---	0; 129
—	45	61	31	0; 130
^	94	126	---	0; 131
[91	123	27	(0; 27)
]	93	125	29	(0; 43)
:	58	42	---	(0; 40)
;	59	43	---	0; 39
@	64	96	---	(0; 26)
,	44	60	---	(0; 51)
.	46	62	---	(0; 52)
/	47	63	---	(0; 53)
¥	92	124	28	---

	92	95	28	---
Enter (数値キー)	13 (224 ; 13)	13 (224 ; 13)	10 (224 ; 13)	(0 ; 166)
 (数値キー)	47 (224 ; 47)	47 (224 ; 47)	(0 ; 149)	(0 ; 164)
* (数値キー)	42	42	(0 ; 150)	(0 ; 55)
- (数値キー)	45	45	(0 ; 142)	(0 ; 74)
+ (数値キー)	43	43	(0 ; 144)	(0 ; 78)
5 (数値キー)	(0 ; 76)	53	(0 ; 143)	---

Appendix A-9 キーボード・コネクタ

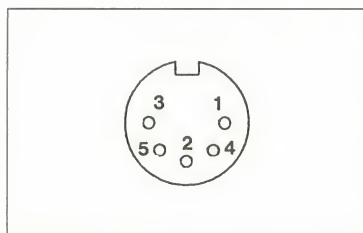
キーボード・コネクタには、6 ピン小型 DIN コネクタと 5 ピン DIN コネクタがあります。どちらのコネクタが使用されるかは、システムごとに異なります。

・ 6 ピン小型 DIN コネクタ



ピン番号	I/O	信号名
1	I/O	データ (DATA)
2		予約
3		グラウンド
4		+5V DC
5	I/O	クロック (CLK)
6		予約

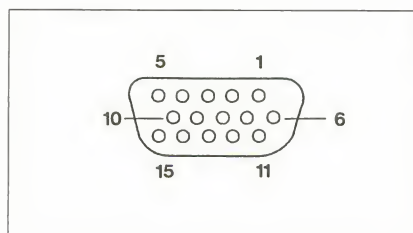
・ 5 ピン DIN コネクタ



ピン番号	I/O	信号名
1	I/O	クロック (CLK)
2		データ (DATA)
3		予約
4		グラウンド
5		+5V DC

Appendix A-10 ディスプレイ・コネクタ

ディスプレイ・コネクタはVGA用15ピン・メス型コネクタです。

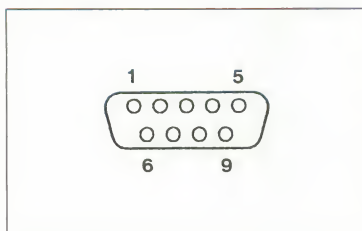


ピン番号	I/O	信号名
1	O	赤色ビデオ信号
2	O	緑色ビデオ信号
3	O	青色ビデオ信号
4		予約
5		グラウンド
6		赤色グラウンド(アナログ)
7		緑色グラウンド(アナログ)
8		青色グラウンド(アナログ)
9		NC
10		グラウンド
11		予約
12		予約
13	O	水平同期信号
14	O	垂直同期信号
15		予約

Appendix A-11 シリアル・ポート・コネクタ

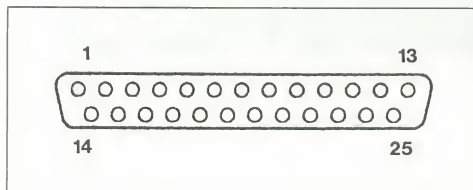
シリアル・ポート・コネクタは、EIA・RS-232C に準拠した 9 ピンと 25 ピンの D シェル・オス型コネクタです。電圧レベルは EIA 規格だけで、電源ループ・インタフェースはサポートされていません。

・ 9 ピン・D シェル・コネクタ



ピン番号	I/O	信号名
1	I	受信キャリア検出
2	I	受信データ
3	O	送信データ
4	O	データ端末レディ
5		信号グラウンド
6	I	データセットレディ
7	O	送信要求
8	I	送信可
9	I	被呼表示

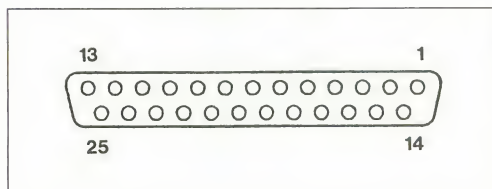
・ 25 ピン・D シェル・コネクタ



ピン番号	I/O	信号名
1		NC
2	O	送信データ
3	I	受信データ
4	O	送信要求
5	I	送信可
6	I	データセットレディ
7		信号グランド
8	I	受信キャリア検出
9		NC
10		NC
11		NC
12		NC
13		NC
14		NC
15		NC
16		NC
17		NC
18		NC
19		NC
20	O	データ端末レディ
21		NC
22	I	被呼表示
23		NC
24		NC
25		NC

Appendix A-12 パラレル・ポート・コネクタ

・ 25 ピン・D シェル・コネクタ



ピン番号	I/O	信号名
1	I/O	ストローブ(不論理)
2	I/O	PD0
3	I/O	PD1
4	I/O	PD2
5	I/O	PD3
6	I/O	PD4
7	I/O	PD5
8	I/O	PD6
9	I/O	PD7
10	I	アクノリッジ(不論理)
11	I	ビジー
12	I	紙なし
13	I	セレクト
14	O	自動紙送り(不論理)
15	I	アラーム(不論理)
16	O	初期化(不論理)
17	O	未使用
18		信号グラウンド
19		信号グラウンド
20		信号グラウンド
21		信号グラウンド
22		信号グラウンド
23		信号グラウンド
24		信号グラウンド
25		信号グラウンド

Appendix A-13 ISA バスの信号位置

・リア・パネル側

—	GND	B 1	A 1	$\overline{\text{I/OCHCK}}$	I
O	RESETDRV	B 2	A 2	SD ₇	I/O
電源	+5 V DC	B 3	A 3	SD ₆	I/O
I	IRQ ₉ (IRQ ₂)	B 4	A 4	SD ₅	I/O
電源	−5 V DC	B 5	A 5	SD ₄	I/O
I	DRQ ₂	B 6	A 6	SD ₃	I/O
電源	−12 V DC	B 7	A 7	SD ₂	I/O
I	$\overline{\text{OWS}}$	B 8	A 8	SD ₁	I/O
電源	+12 V DC	B 9	A 9	SD ₀	I/O
—	GND	B 10	A 10	$\overline{\text{I/OCHRDI}}$	I
O	$\overline{\text{SMEMW}}$	B 11	A 11	AEN	I/O
O	$\overline{\text{SMEMR}}$	B 12	A 12	SA ₁₉	I/O
I/O	$\overline{\text{IOW}}$	B 13	A 13	SA ₁₈	I/O
I/O	$\overline{\text{IOR}}$	B 14	A 14	SA ₁₇	I/O
O	$\overline{\text{DACK}}_3$	B 15	A 15	SA ₁₆	I/O
I	DRQ ₃	B 16	A 16	SA ₁₅	I/O
O	$\overline{\text{DACK}}_1$	B 17	A 17	SA ₁₄	I/O
I	DRQ ₁	B 18	A 18	SA ₁₃	I/O
I/O	$\overline{\text{REFRESH}}$	B 19	A 19	SA ₁₂	I/O
O	SYSCLK	B 20	A 20	SA ₁₁	I/O
I	IRQ ₇	B 21	A 21	SA ₁₀	I/O
I	IRQ ₆	B 22	A 22	SA ₉	I/O
I	IRQ ₅	B 23	A 23	SA ₈	I/O
I	IRQ ₄	B 24	A 24	SA ₇	I/O
I	IRQ ₃	B 25	A 25	SA ₆	I/O
O	$\overline{\text{DACK}}_2$	B 26	A 26	SA ₅	I/O
O	TC	B 27	A 27	SA ₄	I/O
O	BALE	B 28	A 28	SA ₃	I/O
電源	+5 V DC	B 29	A 29	SA ₂	I/O
O	OSC	B 30	A 30	SA ₁	I/O
—	GND	B 31	A 31	SA ₀	I/O

・フロント・パネル側

I	MEMSC 16	D 1	C 1	SBHE	I/O
I	I/OSC 16	D 2	C 2	LA ₂₃	I/O
I	IRQ ₁₀	D 3	C 3	LA ₂₂	I/O
I	IRQ ₁₁	D 4	C 4	LA ₂₁	I/O
I	IRQ ₁₂	D 5	C 5	LA ₂₀	I/O
I	IRQ ₁₅	D 6	C 6	LA ₁₉	I/O
I	IRQ ₁₄	D 7	C 7	LA ₁₈	I/O
O	DACK ₀	D 8	C 8	LA ₁₇	I/O
I	DRQ ₀	D 9	C 9	MEMR	I/O
O	DACK ₅	D 10	C 10	MEMW	I/O
I	DRQ ₅	D 11	C 11	SD ₀₈	I/O
O	DACK ₆	D 12	C 12	SD ₀₉	I/O
I	DRQ ₆	D 13	C 13	SD ₁₀	I/O
O	DACK ₇	D 14	C 14	SD ₁₁	I/O
I	DRQ ₇	D 15	C 15	SD ₁₂	I/O
電源	+5 V DC	D 16	C 16	SD ₁₃	I/O
I	MASTER	D 17	C 17	SD ₁₄	I/O
—	GND	D 18	C 18	SD ₁₅	I/O

Appendix A-14 漢字コード表

シフト	+	1 1 1 1 1 1 1 1 1 1 1 1 1														
JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8140	、	。	。	。	。	。	。	。	。	。	。	。	。	。	。	。
8160	～		
8180	÷	=	<	>	≤	≥	∞	∴	∴	∴	∴	∴	∴	∴	∴	∴
81A0	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	=	ε	ε	ε
81C0																
81E0	≠	<	>	√	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
8240																
8260	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
8280	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
82A0	あ	い	う	え	お	か	き	く	け	こ	さ	し	じ	す	ず	ぜ
82C0	ち	つ	づ	て	と	ど	な	ぬ	ね	の	は	ば	び	び	ふ	ぶ
82E0	も	や	ゆ	よ	ら	り	る	れ	ろ	わ	わ	る	ゑ	を	ん	
8340	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ	タ
8360	チ	ツ	ヅ	テ	ト	ナ	ニ	ノ	ハ	バ	ビ	ビ	フ	ブ	ヘ	ベ
8380	ム	メ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ヰ	ヱ	ヰ	ヱ	ヰ
83A0	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
83C0	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	ρ	σ
83E0																
8440	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
8460	Я															
8480	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
84A0		Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г
84C0																
84E0																
8540																
8560																
8580																
85A0																
85C0																
85E0																
8640																
8660																
8680																
86A0																
86C0																
86E0																
8740																
8760																
8780																
87A0																
87C0																
87E0																

*1: ㄣ

*2: ㄣ

シフト	+	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																														
JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8840	啞	娃	阿	哀	愛	挨	挨	逢	葵	菡	藕	惡	握	渥	旭	葦	芦	葦	葦	葦	葦	葦	葦	葦	葦	葦	葦	葦	葦	葦	葦	葦
8860	安	庵	按	暗	案	闇	較	杏	以	伊	位	依	偉	困	夷	委	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃	萎	衣
8880	謂	違	違	醫	井	亥	域	育	郁	穉	一	壹	從	逸	稻	茨	芋	觸	允	咽	咽	咽	咽	咽	咽	咽	咽	咽	咽	咽	咽	咽
8940	院	陰	隱	頤	吋	右	字	烏	羽	迂	雨	卯	鸛	窺	丑	確	白	渦	噓	嘔	蔚	餓	饒	既	浦	瓜	聞	噲	云	運	雲	
8960	往	餌	載	營	嬰	影	映	曳	榮	永	泳	洩	瑛	盈	穎	穎	英	銜	詠	銳	液	疫	益	馱	謁	越	閱	覆	厭	円		
8980	園	堰	奄	延	怨	掩	援	沿	炎	焰	煙	燕	綠	綠	莖	菌	遠	鈴	鴛	塩	於	汚	甥	凹	央	奧	往	応	押			
89A0	旺	橫	敗	段	王	翁	模	陽	黃	岡	沖	荻	億	屋	億	廬	桶	杜	乙	俺	卸	恩	溫	穩	音	下	化	何	伽	伽		
89C0	佳	加	可	嘉	夏	嫁	寡	寡	暇	果	架	歌	河	火	珂	禍	禾	稼	箇	花	荷	荷	荷	荷	荷	荷	荷	荷	荷	荷	荷	
89E0	霞	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	
8AA0	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶	梶
8AC0	柿	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	鈎	
8AE0	機	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦
8B40	機	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦	婦
8B60	義	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼
8B80	朽	求	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲	汲
8BA0	俠	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑	僑
8BC0	鑾	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞	鸞
8BE0	金	鈴	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀	銀
8C40	形	征	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患
8C60	形	征	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患	患
8C80	劇	戰	擊	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激	激
8CA0	樵	率	犬	獻	研	硯	絹	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩	肩
8CC0	限	乎	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個	個
8CE0	吳	吾	娘	後	御	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧	梧
8D40	后	喉	坑	垢	好	孔	孝	宏	工	巧	巷	幸	庚	庚	弘	恒	慌	抗	拘	控	攻	昂	昂	昂	昂	昂	昂	昂	昂	昂	昂	昂
8D60	港	溝	甲	皇	稷	穡	蘇	紅	紱	絞	網	耕	考	背	肱	腔	膏	航	荒	衡	講	貢	賄	郊	醇	醇	醇	醇	醇	醇	醇	醇
8D80	頃	今	困	坤	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘	壘
8DA0	座	挫	憤	惱	再	最	哉	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞	塞
8DC0	財	呀	坂	阪	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺	堺
8DE0	察	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈	撈
8E40	餐	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫	暫
8E60	死	尸	獅	社	私	私	私	紙	紫	脂	至	視	詞	詩	試	誌	諮	資	賜	離	齒	齒	齒	齒	齒	齒	齒	齒	齒	齒	齒	齒
8E80	滋	治	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾	爾
8EC0	實	郁	篠	櫻	柴	芝	厦	稿	含	射	拾	拾	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	斜	
8EE0	錫	若	叔	弱	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹	惹		

*A: 鯨 / 鯨

*B: 鳶 / 鳶

*C: 鰻 / 鰻

*D: 攪 / 攪

*E: 鳶 / 鳶

*F: 漣 / 漣

*G: 諫 / 諫

*W: 堯 / 堯

*H: 頸 / 頸

*I: 礦 / 礦

*J: 葉 / 葉

シフト	+	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
9040	拭	殖	殖	織	職	色	蝕	蝕	辱	尻	伸	信	侵	唇	販	販	審	心	慎	振	新	晉	森	榛	浸	深	申	疹	真	神			
9060	秦	紳	臣	芯	薪	親	診	身	辛	進	針	震	人	仁	亶	壬	尋	甚	尽	腎	訊	迅	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	*K: 靛 / 粉
9080	逗	吹	垂	帥	推	水	炊	睡	粹	翠	衰	醉	錐	錐	隨	瑞	髓	嵩	數	矩	趨	據	杉	柑	苦	頗	雀	裾	澄				
90A0	摺	忒	世	瀨	欽	是	凄	制	勢	姓	征	性	成	政	整	星	晴	棲	栖	正	清	性	生	盛	精	聖	聖	製	西	誠	誓	請	
90C0	逝	醒	青	靜	齊	稅	喪	雙	席	惜	戚	斥	昔	析	石	積	籍	績	脊	實	赤	跡	蹟	頃	切	拙	接	撰	折	設	窃	節	
90E0	說	雪	絕	舌	蟬	仙	先	千	占	宣	專	尖	川	戰	扇	撲	栓	梅	泉	淺	洗	染	潛	煎	燭	旋	穿	箭	線				
9140	織	羨	腺	舛	船	薦	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	*L: 賤 / 賤	
9160	疏	疎	礎	礎	祖	祖	祖	素	組	蘇	訴	阻	週	鼠	僧	創	双	叢	倉	喪	壯	爽	爽	宋	層	匠	惣	惣	惣	惣	惣	惣	
9180	操	早	曹	巢	槍	槽	槽	槽	燥	爭	瘦	相	窓	精	綜	綜	草	莊	蒼	藻	走	送	道	鎗	霜	露	露	露	露	露	露		
91A0	歲	贈	造	促	側	則	則	息	捉	束	測	足	速	俗	屬	賊	族	統	卒	袖	其	撤	存	孫	損	村	邏	他	多	汰			
91C0	訖	唾	妥	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	恠	
91E0	隊	黨	鋼	代	台	大	第	醒	題	鷹	滝	渾	卓	啄	宅	托	挾	拓	沢	涿	託	錚	濁	諾	茸	鳳	蜻	只					
9240	叩	但	達	辰	奪	脫	異	堅	丕	棚	谷	狙	鯉	樺	誰	丹	卑	嘆	坦	担	担	旦	歎	炭	炭	短	錐	錐	錐	錐	錐	錐	
9260	蛋	誕	鍛	田	壇	彈	彈	暖	暖	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	段	
9280	逐	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭	迭
92A0	帳	巾	巾	張	徵	挑	暢	朝	潮	膝	可	眺	聰	腸	腸	調	調	超	跳	跳	跳	跳	跳	跳	跳	跳	跳	跳	跳	跳	跳	跳	
92C0	搬	鎮	陳	津	璽	椎	趙	追	追	痛	通	塚	梅	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	楓	*M: 壺 / 壺	
92E0	亭	低	停	停	刺	貞	呈	提	定	帝	底	庭	廷	弟	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	抵	
9340	邸	邸	泥	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	掘	
9360	伝	殿	田	電	兎	吐	堵	塗	妬	屠	徒	斗	杜	渡	登	苑	賭	途	都	鍍	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	*N: 礪 / 礪	
9380	凍	刀	唐	塔	塔	套	岩	島	嶋	埒	投	搭	東	桃	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	*O: 礪 / 礪	
93A0	蕩	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	葎	*P: 濤 / 濤
93C0	流	特	督	禿	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	驚	
93E0	如	乍	風	蓬	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	謎	*Q: 瀝 / 瀝
9440	如	尿	菲	任	妊	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	忍	
9460	現	蚤	巴	把	播	霸	把	波	派	破	婆	罵	告	局	俳	庖	排	排	敗	杯	盃	背	背	背	背	背	背	背	背	背	背	背	
9480	煤	煤	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	買	*R: 蠅 / 蠅
94A0	箱	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕	裕
94C0	步	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫
94E0	扉	批	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	披	
9540	鼻	痔	痔	匹	正	訖	彥	膝	麥	肘	弼	必	畢	筆	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	𠂔	
9560	描	病	秒	苗	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸	鋸
9580	斧	昔	浮	付	符	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣	廣
95A0	腹	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複	複
95C0	並	蔽	閉	陸	米	頁	僻	壁	碧	碧	碧	別	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡	賡
95E0	步	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫	甫
9640	法	泡	烹	炮	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍	龍
9660	冒	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪	訪
9680	摩	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨	磨
96A0	迷	銘	鳴	姪	化	減	免	棉	綿	綿	綿	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	*T: 襪 / 襪
96C0	迷	銘	鳴	姪	化	減	免	棉	綿	綿	綿	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵	麵
96E0	初	貫	問	問	紋	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門	門
9740	論	唯	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑	佑
9760	與	預	備	幼	妖	容	席	揚	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	擁	*Y: 遙 / 遙
9780	沃	浴	翌	翼	淀	羅	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺	螺
97A0	裏	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡	裡
97C0	梁	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼	涼
97E0	伶	例	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷	冷

*K: 軀/粉

*L: 賤/賤

*M: 壺/壺

*N: 礪/礪

*O: 礪/礪

*P: 濤/濤

*Q: 逕/逕

*R: 蠅/蠅

*S: 檜/檜

*X: 檜/檜

*T: 檜/檜

*U: 蕨/蕨

*Y: 遙/遙

[illegible]

シフト	+	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1															
JIS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	F
E040	濛	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓
E060	濛	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓	漓
E080	烙	焉	煨	煨	煨	煨	煨	煨	煨	煨	煨	煨	煨	煨	煨	煨	煨
E0A0	燿	燿	燿	燿	燿	燿	燿	燿	燿	燿	燿	燿	燿	燿	燿	燿	燿
E0C0	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊
E0E0	珥	珥	珥	珥	珥	珥	珥	珥	珥	珥	珥	珥	珥	珥	珥	珥	珥
E140	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠	瓠
E160	畫	略	略	略	略	略	略	略	略	略	略	略	略	略	略	略	略
E180	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡	瘡
E1A0	癩	癩	癩	癩	癩	癩	癩	癩	癩	癩	癩	癩	癩	癩	癩	癩	癩
E1C0	昵	眈	眈	眈	眈	眈	眈	眈	眈	眈	眈	眈	眈	眈	眈	眈	眈
E1E0	矜	矜	矜	矜	矜	矜	矜	矜	矜	矜	矜	矜	矜	矜	矜	矜	矜
E240	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉
E260	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉
E280	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉	穉
E2A0	并	并	并	并	并	并	并	并	并	并	并	并	并	并	并	并	并
E2C0	篋	篋	篋	篋	篋	篋	篋	篋	篋	篋	篋	篋	篋	篋	篋	篋	篋
E2E0	杵	杵	杵	杵	杵	杵	杵	杵	杵	杵	杵	杵	杵	杵	杵	杵	杵
E340	紂	紂	紂	紂	紂	紂	紂	紂	紂	紂	紂	紂	紂	紂	紂	紂	紂
E360	總	總	總	總	總	總	總	總	總	總	總	總	總	總	總	總	總
E380	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
E3A0	疊	疊	疊	疊	疊	疊	疊	疊	疊	疊	疊	疊	疊	疊	疊	疊	疊
E3C0	翅	翅	翅	翅	翅	翅	翅	翅	翅	翅	翅	翅	翅	翅	翅	翅	翅
E3E0	聰	聰	聰	聰	聰	聰	聰	聰	聰	聰	聰	聰	聰	聰	聰	聰	聰
E440	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍
E460	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍
E480	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍
E4A0	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖
E4C0	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖
E4E0	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖	莖
E540	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻
E560	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻	蕻
E580	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
E5A0	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
E5C0	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
E5E0	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
E640	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦
E660	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼	誼
E680	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱
E6A0	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱	諱
E6C0	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤
E6E0	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤
E740	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤
E760	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤
E780	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤
E7A0	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤
E7C0	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤	賤
E7E0	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞

*Z:瑞/瑤

*i:砥/磧

*n:砺/礪

*c:甍/甍

*v:甍/甍

*j:蕊/藥

*u:藪/藪

*c:蛭/蛭

*r:蠅/蠅

*g:諫/諫

*l:賤/賤

*q:迺/迺

[illegible]

INDEX

— 記号 —	
\$DISP.SYS	31
\$FONT.SYS	30
\$IAS.SYS	32
\$PRNESC.P.SYS	33

— A —	
APTO	25
AT バス	20

— B —	
BIOS コール	37

— D —	
DBCS	39

— E —	
EGA	25
ESC/P J84	55

— J —	
JIS-X0208	43

— M —	
MCA	21

— O —	
OADG	26

— V —	
V-Text	58

VGA	25
-----------	----

— W —	
Windows	62

— え —	
エスケープ・シーケンス	47

— お —	
オープン・アーキテクチャ	18

— か —	
カーソル形状	46
外字領域	43

— き —	
キー・バッファ	50
疑似ビデオ・バッファ	45

— こ —	
高品位テキスト・モード	59
高密度テキスト・モード	59
コード・ページ	42

— し —	
システム・コール	37
システム予約領域	46
シフト JIS コード	38

— せ —	
選定文字	43

— そ —		入力支援ドライバ32	
走査コード	49	— ひ —	
— て —		ビデオ・モード	44
ディスプレイ・ドライバ	37	— ふ —	
— に —		ファンクション・コール	37
日本語 FEP	50	フォント・ドライバ	30
日本語入力 FEP	32	プリンタ・ドライバ	33

DOS/V テクニカル・リファレンス・マニュアル

1993 年 9 月 25 日 初版発行

著 者………^{あ だち つよし}芦達 剛

発行者………橋本 五郎

発行所………ソフトバンク株式会社 出版事業部

〒103 東京都中央区日本橋浜町 3-42-3

販売 03(5642)8101

編集 03(5642)8143

制作・印刷………有限会社バリエ社

装 丁………花本 浩一

落丁本、乱丁本は小社販売局にてお取り替え致します。

定価はカバーに記載されています。

Printed in Japan

ISBN 4-89052-440-1

◆ソフトバンクのC言語の本◆

C言語 プログラミングの エッセンス

結城浩 著

A5判・344ページ 定価2,900円



Cマガジンの連載記事の書籍化。考えるための様々なヒント（エッセンス）を分かりやすく明快に解説。サンプルプログラムはPC9800シリーズ、DOS/Vマシン、FM TOWNS に対応。

Cプログラマのための 新MS-DOS プログラミング入門

中島信行 著

B5判・248ページ 定価3,300円



C MAGAZINE連載の「Cプログラマのための新MS-DOSプログラミング入門」を書籍化。「MS-DOSの機能を拡張するためのCプログラミング技法」の続編。

秘伝C言語問答 ポインタ編

柴田望洋 著

B5判・303ページ 定価2,600円

C言語を修得する上で最大の難関とされるポインタについて、先生と生徒との問答形式でやさしく解説しています。難しいポインタの概念を初心者にもわかりやすく解説した一冊です。

C言語による 実践MS-DOS プログラミング入門

秋津彰文 著

A5判・280ページ 定価2,200円

Cプログラミング上で必須なMS-DOSの知識を提示し、C言語とMS-DOSの関係を活かしたサンプルプログラムを多数掲載しています。

C:98 スーパーライブラリ

柴田望洋 著

A5判・378ページ 定価3,700円



グラフィック、ポップアップウィンドウなど豊富な機能を持ち、日本で発売されているほとんどの処理系に対応しているC言語のライブラリです。付属ディスク（2HD）に全ソースを収録。

入門 Turbo C++

立野繁之・武田和宏 共著

A5判・392ページ 定価2,900円

Turbo C++を初めて使うユーザーのための入門書です。インストール、統合環境やコマンドラインの操作、さらにはライブラリリファレンスまで、例題を挙げながら解説しています。

Effective C++

S.マイヤーズ 著 岩谷宏 訳

B5判・232ページ
定価3,200円

Addison Wesley プロフェッショナル・コンピューティングシリーズ第1弾。C++の実践上の主要な問題点を学びやすく印象に残りやすいように、切りのよい50カ条にまとめた、C++の解説書として最も優れた1冊。

Cの実験室 初級ラボ編/中級ラボ編

B5変型判・2色刷り 208ページ/224ページ
林晴比古 著 初級編-定価1,600円/中級編-定価1,800円

C言語のあらゆるルールを、実験や紙上シュミレーションを重ねることで簡単に理解していける画期的な入門書。初心者からマニアまで満足のいく内容です。初級ラボ編には77、中級ラボ編には69の実験例を収録。

Cプログラマのための C++入門

柴田望洋 著

B5変型判・340ページ 定価2,900円

俊英 柴田望洋が贈る、C++解説書の決定版。CからC++へ、その拡張の必然性とC++の特長を系統的に解説しています。

MS-DOS ポータブル プログラミング

中島信行 著

B5判・168ページ 定価2,900円



C言語でポータブルなプログラムを作成するための方法を、多くの資料と具体的なソースリストを通して、分かりやすく解説。

Cのオモチャ箱

Amuse yourself with C Programming

植村富士夫 著

B5変型判・232ページ 定価2,900円



実用的で応用のきくプログラム10個をもとに、プログラミングの楽しさと醍醐味、実際にプログラムを書くためのノウハウを紹介。プログラムにはそれぞれの作成方法、使用法などを付けました。

◆ ソフトバンクのC言語の本 ◆

新C言語入門 スーパービギナー編

林晴比古 著

B5変型判・212ページ 定価1,500円

Cプログラマー一年生に贈る待望の入門書。ビジュアルを駆使した解説で、「Cのコツ」を誰でも身につけることができます。いきなりC言語にチャレンジしたい超初心者には決定版の一冊です。

新C言語入門 ビギナー編

林晴比古 著

B5変型判・256ページ 定価1,900円

C言語実用マスターシリーズ第1巻。複雑なC言語の知識を整理、要約しました。数多くのサンプルプログラムを駆使して「誰にでもわかるC」を実現。ANSI標準規格にも完全対応しています。

新C言語入門 シニア編

林晴比古 著

B5変型判・400ページ 定価2,400円

簡単なプログラムなら作成できるという方に、C言語の仕様を体系的に解説。難しい概念はわかりやすく図表化して理解を促していきます。Cの思想そのものについて解き明かした本格的解説書です。

新C言語入門 応用編

林晴比古 著

B5変型判・366ページ 定価2,400円

C言語実用マスターシリーズ完結編。実用的なプログラムを自分で作成してみたいという方を対象に、C言語のさまざまな応用技術を紹介。実例プログラムを用いて、詳しく解説しました。

Cプログラマのための アルゴリズムとデータ構造

近藤嘉雪 著

A5判・244ページ 定価2,200円

代表的なアルゴリズムを紹介しながら、その実例をCプログラムで実現しています。Cプログラマー中・上級者の方へおすすめの一冊。

プラムの実践C プログラミング講座

B5判

332ページ

Thomas Plum 著 福富寛・曽根秀昭 訳 定価3,400円

米国のC言語セミナーのテキストを書籍化。プロのCプログラマーを目指す人達に、UNIX Cの特長と修得のコツを解説しています。

Turbo C++の応用50例

中山雅彦・井上俊宏 共著

A5判・304ページ 定価3,200円

さまざまなジャンルのプログラムを満載し、明快に解説。Borland C++やTurbo C Ver.2.0にも対応。付録ディスクには、全ソースに加え、Turbo C Ver.2.0用ソース、実行ファイルを収録しています。

C++ Cプログラマのための実践ガイド

B5判・285ページ

S.ヘクマトプール 著 岩谷宏 訳

定価3,600円

Cをある程度理解した方に、C++の具体的な機能を実例をあげて紹介。C++による本格的な開発の事例をわかりやすく解説。

STANDARD C

P.J.プラウガー & J.プロディ 著 福富寛ほか 訳

A5判・344ページ 定価3,200円

コンピュータ言語の研究者として世界的に著名なP.J.プラウガー氏の書き下ろしによる、「ANSI C」の教科書ともいえる一冊。ANSI標準Cの特徴を、わかりやすく解説しました。

Perl プログラミング

Larry Wall & Randal L.Schwartz 著 近藤嘉雪 訳

A5判・640ページ 定価4,500円

現在圧倒的な人気と信頼を誇る、米国O'Reilly社のNUTSHELLハンドブックシリーズの邦訳版。perlの作者自らがユーモアあふれる文章で、明快に解説。真正正銘のperlの定本です。

Practical C Programming 現実的なCプログラミング

Steve Oualline 著 岩谷宏 訳

A5判・513ページ

定価3,600円

MS-DOSユーザとUNIXユーザのためのC言語の本格的入門書。仕様決定からデバッグ・完成・発表・改版まで、プログラミングの全課程にわたり解説。

the 1990s, the number of people in the UK with a long-term condition has increased by 50% (Department of Health 2000). The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

Long-term conditions are a major cause of disability and are a leading cause of death in the UK. The prevalence of long-term conditions is increasing in the UK and in other countries. The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

The prevalence of long-term conditions is increasing in the UK and in other countries. The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

The prevalence of long-term conditions is increasing in the UK and in other countries. The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

The prevalence of long-term conditions is increasing in the UK and in other countries. The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

The prevalence of long-term conditions is increasing in the UK and in other countries. The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

The prevalence of long-term conditions is increasing in the UK and in other countries. The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

The prevalence of long-term conditions is increasing in the UK and in other countries. The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

The prevalence of long-term conditions is increasing in the UK and in other countries. The prevalence of long-term conditions is also increasing in other countries (e.g. Australia, Canada, France, Germany, Italy, Japan, the Netherlands, New Zealand, Norway, Sweden, Switzerland, Taiwan, the USA and the USSR) (Department of Health 2000).

郵便はがき

料金受取人払

1 0 3 - 0 0

日本橋局
承認

1 6 1

1565

差出有効期間
平成7年4月
30日まで

東京都中央区

日本橋浜町3-42-3

ソフトバンク株式会社 出版事業部
パソコン言語 書籍編集部 行

住所	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	☎			
氏名						年齢	性別	男女	
職業・勤務先 学校(学部)・学年						所有機種			

愛読者アンケート

お買上げ
の 書 名

DOS/V テクニカル・リファレンス・マニュアル

○お読みになられた感想をお聞かせください

○お買上げの動機をお聞かせください

○これからどんな本をご希望ですか

○主にパソコンをどのような目的にお使いですか

○C 言語の習熟度をお聞かせください

- ①これから学習 ②文法を理解できる ③小型のツールを組める
④中型アプリケーションを組める ⑤システムの記述ができる

○ご購入の新聞・雑誌

新聞名ー

雑誌名ー

○本書をお買い上げの書店名

都・道
府・県

市・区

書店

アンケートにご協力ありがとうございました。
今後とも、小社出版物をよろしく願います。



ソフトバンク

ISBN4-89052-440-1

C0055 P3200E



9784890524402

定価3,200円

(本体3,107円)



1910055032002

DOS/V

DOS/V

テクニカル

Technical

リファレンス

Reference

マニュアル

Manual